

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

**BAKALÁŘSKÁ PRÁCE**

2010

Michal Dojcsán

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra měřicí a řídicí techniky**

**Řízení pohybu laserového ukazovátka**  
**Motion control of laser pointer**

# Zadání bakalářské práce

Student: **Michal Dojcsán**

Studijní program: B2645 Elektrotechnika, sdělovací a výpočetní technika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: **Řízení pohybu laserového ukazovátka**  
**Motion control of laser pointer**

Zásady pro vypracování:

1. Seznámení s principy diskrétní Fourierovy transformace
2. Návrh a realizace systému pro ovládání pohybu projekční laserové diody
3. Test systému s dynamickými obrazy vytvořenými pomocí DFT
4. Test systému se statickými bitmapami
5. Zhodnocení dosažených výsledků

Seznam doporučené odborné literatury:


- [1] Pirk J.: Řešené příklady v C#. KOPP, České Budějovice 2005. ISBN 80-7232-265-6
- [2] Virius M.: C# Hotová řešení. Computer Press, Praha 2006. ISBN 80-251-1084-2
- [3] Vít V.: Televizní technika 4a. BEN - technická literatura, Praha 2002. ISBN 80-86056-74-0
- [4] Zonst A.E.: Understanding FFT Applications. Citrus press 2003. ISBN-10: 0964568144

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 30.11.2008

Datum odevzdání: 07.05.2010



doc. Ing. Jiří Koziolek, Ph.D.  
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum odevzdání bakalářské práce: 7.5.2010 .....

Děkuji panu Ing. Zdeňkovi Slaninovi, Ph.D za odbornou pomoc a konzultace při  
zpracování bakalářské práce

## **Abstrakt**

### **Česky:**

Tato práce se zabývá návrhem a realizací systému pro řízení laserového paprsku k vytvoření statických a dynamických obrazců reagujících na frekvenční spektrum, získané diskrétní Fourierovu transformací. To je analýza k určení frekvenčních složek, ze kterých se skládá analyzovaný signál. Výsledkem analýzy je amplituda jednotlivých složek, což umožňuje z naměřených dat příslušně reagovat a měnit tvar a velikost obrazců promítaných na projekční plochu. Podle toho jaké frekvence se v analyzovaném signálu nachází, vzniká laserovým paprskem vytvořený obrazec, který má jedinečný tvar pro vstupní signál. Rozvoj výpočetních technologií zrychluje výpočet algoritmů frekvenčního spektra signálu v reálném čase a to umožňuje například vizualizaci signálu ve frekvenční oblasti. Využití se nabízí ve vědeckých měřeních i v komerční elektronice.

### **English:**

This bachelor work is concerned about the concept and realization of ray control system to make static and dynamic patterns that react on frequency spectrum that is gained by the Fourier's transformation. This is the analysis to establish frequency components from which the analysed signal is composed of. The analysis outcome is the amplitude of individual components that enables from the measured data to respectively react and change shape and size of the patterns projected on design area. Depends on what kind of frequencies are in the analysed signal, a new pattern is being made by the laser ray that has its individual shape for the input signal. The development of computational technology accelerates the calculation a algorithms of frequency signal spectrum in a real time and this e.g. enables visualization of the signal in the frequent scope. The utilising is possible in scientific measurements and in commercial electronics as well.

## **Klíčová slova**

### **Česky:**

Digitální signálový procesor, DSP56F805, CodeWarrior, Processor Expert, frekvenční analýza, DFT, FFT, amplitudové spektrum, laser, hypotrochoida.

### **English:**

Digital signal processor, DSP56F805, CodeWarrior, Processor Expert, frequency analysis, DFT, FFT, amplitude spectrum, laser, hypotrochoid

## Seznam zkratek

A/D	Analogově digitální
CMOS	Complementary Metal Oxide Semiconductor (technologie výroby)
DSP	Digital Signal Processor (Digitální Signálový Procesor)
DFT	Discrete Fourier transform (Diskrétní Fourierova Transformace)
FFT	Fast Fourier transform (Rychlá Fourierova Transformace)
FPGA	Field-Programmable Gate Array (Programovatelné Hradlové Pole)
GPIO	General Purpose Input/Output (vstupně výstupní brána)
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation (Pulzně Šířková Modulace)
RS232	Recommended Standard 232 (Komunikační Rozhraní)
PE	Processor Expert™

## Seznam symbolů

$f$	frekvence
$f_s$	frekvence vzorkování
$\mathbf{F}\{.\}$	Fourierova transformace $\{.\}$
$\mathbf{F}^{-1}\{.\}$	zpětná Fourierova transformace $\{.\}$
$\mathbf{FR}\{.\}$	rozvoj $\{.\}$ do Fourierovy řady
$\mathbf{FR}^{-1}\{.\}$	součet Fourierovy řady $\{.\}$
$K$	konstanta
$k$	poměr
$m$	index
$N$	počet vzorků
$T$	perioda
$T_s$	perioda vzorkování
$t$	čas
$u[k]$	vstupní signál soustavy s diskretním časem
$\mathbf{w}_m$	komplexní koeficient Fourierovy řady
$w(t)$	reálný signál
$^{\mathbf{FR}}\mathbf{W}$	spektrum signálu
$y[k]$	výstupní signál soustavy s diskretním časem

## **Obsah:**

1. Úvod .....	1
2. Principy a rozbor diskrétní Fourierovy transformace .....	2
2.1 Definice signálu s diskrétním časem .....	2
2.3 Spektrum $^{FR}W$ signálu s diskrétním časem $w[k]$ .....	3
3. Návrh systému pro ovládání laserového paprsku .....	6
3.1 Teorie vzniku geometrických obrazců pomocí obvodů kružnic .....	6
3.1.1 Epicykloida: .....	6
3.1.2 Epitrochoida: .....	7
3.1.3 Hypocykloida: .....	8
3.1.4 Hypotrochoida: .....	9
3.2 Spirograf .....	11
3.3 Návrh mechanického systému pro rozmítání laserového paprsku .....	12
3.3.1 Hlavní princip a konstrukce .....	12
3.3.2 Použité motory: .....	13
3.3.3 Mechanická konstrukce posunovacího zařízení .....	16
3.4 Laserový modul .....	18
3.5 Spínací elektronika výkonových částí .....	19
4. Přepočet parametrů z geometrického systému na veličiny vykreslovací jednotky ....	20
4.1 Převod parametrů obrazců z geometrického na zrcátkový systém .....	21
4.2 Přepočet z parametrů obrazců na říditelné veličiny motorů a servomotorů .....	22
5. Základní problematika DSP .....	25

5.1 Úvod .....	25
5.2 Obecná aplikace .....	25
5.3 Architektura.....	26
5.4 Periférie .....	27
5.5 Programování .....	28
5.6 Rozdělení.....	29
6. Hardware řídicího systému pro vykreslovací jednotku .....	30
6.2 Technické parametry vývojového kitu DSP56F805.....	32
6.3 Výpočet frekvence A/D převodníku.....	33
7. Software řídicího systému pro vykreslovací jednotku .....	34
7.1 Program v DSP .....	34
7.1.1 Statická funkce .....	36
7.1.2 Dynamická funkce.....	39
7.2 Program v PC .....	45
7.1.2 Statická funkce .....	46
7.1.2 Dynamická funkce.....	50
8. Závěr .....	52
8.1 Test se statickými obrázky .....	52
8.1.1 Zhodnocení výsledků statických obrázků.....	55
8.2 Test s dynamickými obrázky.....	56
8.1.1 Zhodnocení výsledků dynamických obrázků.....	56
9. Literatura: .....	57
10. Přílohy: .....	58



## **1. Úvod**

Cílem práce je mimo objasnění principů diskrétní Fourierovy transformace vytvořit systém s ovládáním laserového paprsku tak aby byl schopen vytvářet měnící se obrazce dle měnícího se frekvenčního spektra právě měřeného signálu.

Navržený systém se skládá ze tří částí. Rozmítání paprsku má na starosti mechanický systém se zrcátky, motory a servomotory. Vzorkování a výpočet DFT má za úkol vývojový kit, který je osazen digitálním signálovým procesorem firmy Freescale. Třetí částí je PC s komunikačním rozhraním RS232 a příslušným softwarem, který posílá data do DSP.

Zrcátkový rozmítací systém je schopen vychylovat a projektovat laserový paprsek v horizontálním a vertikálním směru na projekční plochu (zeď, plátno). Vytvářené obrazce jsou výsledkem dvou rotujících zrcátek svírající s osou otáčení proměnlivý úhel. Obrazce, které lze touto metodou vykreslit mají svá přesná označení. Např.: epitrochoida, hypotrochoida, epicykloida, hypocykloida.

Parametry zobrazovacího systému, kterými jsou rychlost otáček motorů, úhly zrcátek a laserový modul, jsou řízeny mikrokontrolerem typu DSP z řady DSC56F8xx. Tento digitální signálový procesor by měl být dostatečně rychlý pro zpracování FFT (fast fourier transform) a k obsluze vstupů a výstupů potřebných periférií.

Uživatelský zásah do systému je umožněn díky jednosměrné komunikaci přes sériovou linku s PC. Aplikace posílá do DSP informace o tom, jak se má vychylovací jednotka chovat, co má zobrazovat a jak má reagovat na frekvenční spektrum. I bez propojení PC s programovacím kitem je systém schopný jednoduchého automatického provozu.

## **2. Principy a rozbor diskrétní Fourierovy transformace**

### **2.1 Definice signálu s diskrétním časem**

Signál s diskrétním časem  $w[k] = w(t_k)$  má hodnotu definovanou pouze v diskrétních časových okamžicích  $t_k$ . signál  $w[k]$  je tvořený časovou posloupností hodnot  $w(t_k)$ . Analogicky impulsnímu signálu zvolíme  $t_k = kT_s$ . Je tedy:

$$w(kT_s) = w[k]$$

Signál s diskrétním časem  $w[k]$  je časová posloupnost hodnot, do které je zakódována informace.

Signál s diskrétním časem je modelem skutečného signálu, volba jeho použití závisí na konkrétních fyzikálních problémech, které jsou předkládány k řešení.

V základním modelu lze předpokládat, že se signál s diskrétním časem převádí tak, že se za hodnoty  $w[k]$  dosadí mohutnost Diracových impulsů v signálu  $w_I(t)$ . Hodnota vzorků vzorku je  $w(kT_s)$ .

$$w[k] = \int_{kT_s^-}^{kT_s^+} w_I(t) dt$$

Aby byly výsledky analýzy signálu s diskrétním časem interpretovatelné, musí vzorkování signálu  $w(t)$  splňovat podmínky Shanonova a Kotělnikova teorému, který říká, že pro obnovu signálu  $w(t)$  musí být frekvence vzorkování  $f_s$  větší, než je dvojnásobek maximální frekvence signálu  $w(t)$ , tedy:

$$f_s > 2f_{\max}$$
$$\omega_s > 2\omega_{\max}$$

Případy, kdy je obecný vstupní signál s diskrétním časem zpracováván soustavou analyticky jsou velmi řídké. Povaha úloh vede k zadáním, která se řeší numericky.

Perioda  $K_A$  periodického signálu  $w_A[k]$  je dána vztahem?

$$\begin{aligned}w_A[k] &= w[k + K_A] \\k &= \dots, -2, -1, 0, 1, 2, \dots \\K_A &\in \mathbb{N}; K_A \geq 1\end{aligned}$$

Lze si všimnout, že odvozujeme-li signál s diskretním časem od signálu harmonického,

$$w_A[k] = w_{A,\max} \cos(\omega_A k T_s)$$

bude signál  $w_A[k]$  periodický pouze tehdy, bude-li  $\omega_A T_s / \pi$  racionální číslo.

Pojem perioda  $K_A$  signálu s diskretním časem  $w_A[k]$  se tedy nedefinuje vzhledem k úhlovému kmitočtu signálu se spojitým časem  $w_A(t)$ , jehož měřením by mohl být signál  $w_A[k]$  generován.

Z hodnoty  $K_A$  nelze určit, nebo odhadnout, na rozdíl od hodnoty periody  $T_A$  signálu  $w_A(t)$ , základní frekvenci vzorkovaného periodického signálu.

Pojem frekvence  $f_A$  nebo úhlový kmitočet  $\omega_A = 2\pi f_A$  signálu  $w_A[k]$  se proto definuje frekvencí nebo úhlovým kmitočtem signálu  $w_A(t)$ , jehož vzorkováním mohl být signál  $w_A[k]$  generován.

[1]

### **2.3 Spektrum <sup>FR</sup>W signálu s diskretním časem $w[k]$**

Fourierovo spektrum signálu  $w[k]$  je definováno pro signál periodický,

$$w_A[k] = w[k + K_A]$$

Kde  $K_A$  je perioda signálu  $w_A[k]$ .

Každý konečný periodický signál  $w[k]$  se základní periodou  $K_A$  lze pro  $k = \dots, -2, -1, 0, 1, 2, \dots$  vyjádřit Fourierovou řadou

$$w_A[k] = \sum_{m=0}^{m=K_A-1} w_m \exp\left(jm \frac{2\pi}{K_A} k\right) = FR^{-1}\{w_m\}$$

Kde komplexní koeficienty  $w_m$  Fourierovy řady pro signál s diskretním časem jsou

$$w_m = \frac{1}{K_A} \sum_{k=0}^{k=K_A-1} w_A[k] \exp\left(-jm \frac{2\pi}{K_A} k\right) = FR\{w_A[k]\}$$

$$m = 0, 1, 2, \dots, K_A - 1$$

Koeficienty  $w_m$  Fourierovy řady jsou komplexní prvky spektra signálu  $w_A[k]$ , jde tedy o

vektory ve spektru signálu  $w(t)$ . 
$$w_m = w_m \exp\left(jm \frac{2\pi}{K_A} k\right)$$

Fourierovy řady  $w_A[k]$  jsou komplexní složky signálu  $w_A[k]$ , jsou to vektory, jejichž součet je signál  $w_A[k]$ .

Fourierova řada  $w_A[k]$  obsahuje konstantní složku a dvojice komplexně sdružených vektorů. Dvojice vektorů se sčítají do reálné složky signálu  $w_A[k]$ . součet složek signálu  $w_A[k]$  je opět signál  $w_A[k]$ .

Spektrum periodického signálu s diskretním časem je teoretická konstrukce, jejímž cílem je vybudovat v kmitočtové oblasti popis signálu s diskretním časem ve tvaru, analogickém popisu signálů se spojitým časem.

Fourierova řada popisuje  $K_A$  hodnot periodického signálu s diskretním časem a hodnotami  $K_A$  periodických ortogonálních signálů s diskretním časem, komplexních exponenciál. Parametry těchto  $K_A$  exponenciál jsou určeny z  $K_A$  hodnot jedné periody periodického signálu  $w_A[k]$ , k redukci dat zde nedochází. Stejně jako u spektra signálu se spojitým časem, exponenty exponenciál charakterizují, s jakou rychlostí se signál  $w_A[k]$  mění v čase.

[1]

Spektrum  ${}^{\text{FR}}\mathbf{W}_A$  periodického signálu  $w_A[k]$  je množina

$${}^{\text{FR}}W_A = \{w_m\}$$

kde:

$$w_m = \frac{1}{K_A} \sum_{k=0}^{K_A-1} w_A[k] \exp\left(-jm \frac{2\pi}{K_A} k\right)$$

Stručně se zapisuje:

$$\{w_m\} = \text{FR}\{w_A[k]\}$$

$$w_A[k] = \text{FR}^{-1}\{w_m\}$$

Spektrum amplitudy  ${}^{\text{FR}}|\mathbf{W}_A|$  periodického signálu  $w_A[k]$  je množina

$${}^{\text{FR}}|W_A| = \{|w_m|\}$$

Spektrum amplitudy  $|W_{(\omega)}|$  signálu  $w[n]$

$$|W_{(\omega)}| = \sqrt{(\text{Re}\{W_{(\omega)}\})^2 + (\text{Im}\{W_{(\omega)}\})^2}$$

Spektrum fáze  ${}^{\text{FR}}\theta_A$  periodického signálu  $w_A[k]$  je množina

$${}^{\text{FR}}\theta_A = \{\theta_m\}$$

$$\text{kde } \theta = \arg(w_m)$$

Spektrum výkonu  ${}^{\text{FR}}P_A$  periodického signálu  $w_A[k]$  je množina

$${}^{\text{FR}}P_A = \{|w_m|^2\}$$

### 3. Návrh systému pro ovládání laserového paprsku

#### 3.1 Teorie vzniku geometrických obrazců pomocí obvodů kružnic

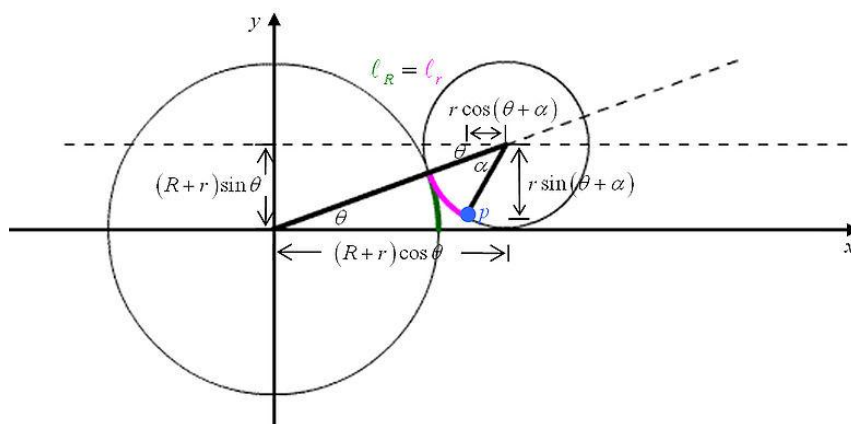
##### 3.1.1 Epicykloida:

Je křivka, která vzniká pohybem bodu, který leží na kružnici  $r$ . Tato kružnice  $r$  se ovaluje kolem statické kružnice  $R$ . Je to zvláštní skupina rotací, kdy má většinou menší kruh poloměr  $r$  a větší kruh poloměr  $R$ ,  $R = k * r$ , kde  $k$  je poměr poloměru  $R$  k poloměru  $r$ . Platí parametrická rovnice:

$$x(\theta) = (R + r) \cdot \cos \theta - r \cdot \cos \left( \frac{R + r}{r} \cdot \theta \right)$$

$$y(\theta) = (R + r) \cdot \sin \theta - r \cdot \sin \left( \frac{R + r}{r} \cdot \theta \right)$$

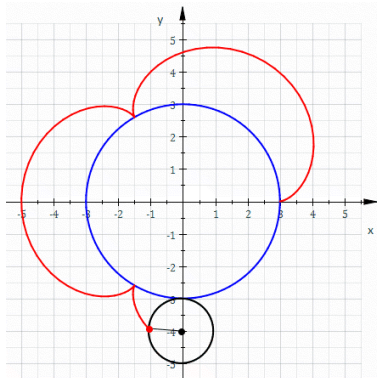
Nákres a popis kružnic v kartézské souřadnici:



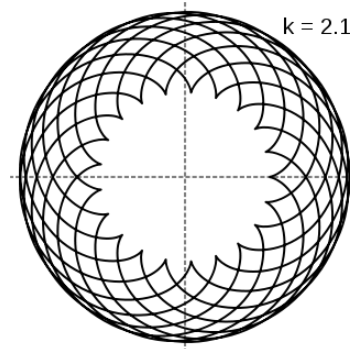
obr.3.1 epicykloid v kartézské souřadnici

[9]

### Příklady epicykloid:



obr.3.2 vznikající epicykloida:  $r = 1$   $R = 3$



obr.3.3  $k = R/r = 19/5 = 2.1$

[8]

### 3.1.2 Epitrochoida:

Je křivka, která vzniká pohybem bodu spojeného s kružnicí s poloměrem  $r$ , která se ovaluje okolo statické kružnice s poloměrem  $R$ . Pohybující se bod je ve vzdálenosti  $d$  od středu menší kružnice s poloměrem  $r$ , která se ovaluje kolem vnitřní statické kružnice. křivku lze

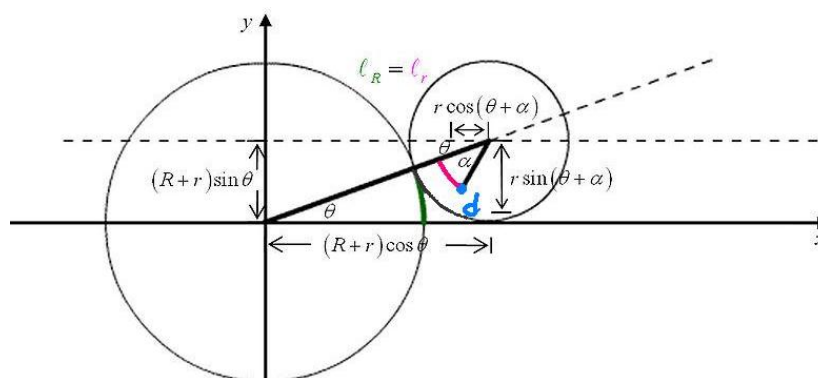
vyjádřit v parametrickém tvaru jako:

$$x(\theta) = (R+r) \cdot \cos \theta - d \cdot \cos \left( \frac{R+r}{r} \cdot \theta \right)$$

$$y(\theta) = (R+r) \cdot \sin \theta - d \cdot \sin \left( \frac{R+r}{r} \cdot \theta \right)$$

kde  $\theta$  je úhel otáčení. Pokud  $d = r$  (bod se nachází přímo na větší kružnici) nazývá se křivka epicykloida. Epitrochoidní tvar má například komora Wankelova motoru.

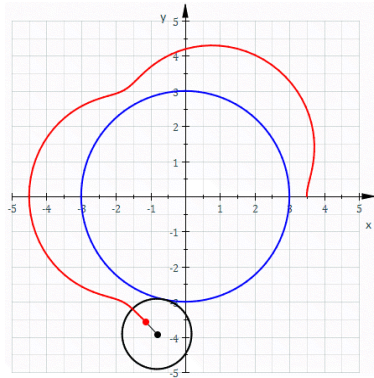
nákres a popis kružnic v kartézské souřadnici:



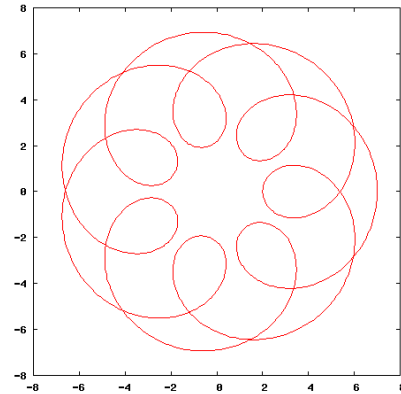
obr.3.4 epitrochoida v kartézské souřadnici

[9]

příklady epitrochoid:



obr.3.5 epitrochoida:  $R = 3$   $r = 1$   $d = 0,5$



obr.3.6  $k = 3,5$   $R=3,5$   $d = 2,5$

[8]

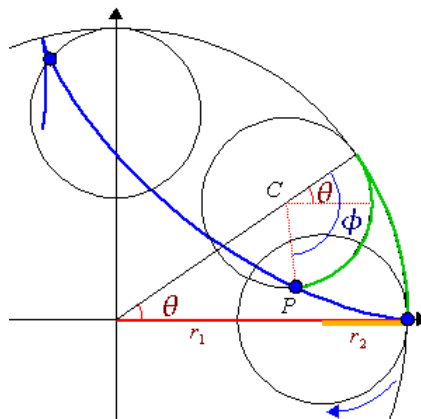
### 3.1.3 Hypocykloida:

Je křivka, která vzniká pohybem bodu, který leží na kružnici  $r$ . Tato kružnice  $r$  se ovaluje podél vnitřního okraje statické kružnice  $R$ . Je to zvláštní skupina rotací, kdy má menší kružnice poloměr  $r$  a větší kružnice poloměr  $R$ ,  $R = k * r$ , kde  $k$  je poměr poloměru  $R$  k poloměru  $r$ . Platí parametrická rovnice:

$$x(\theta) = (R - r) \cdot \cos \theta + r \cdot \cos \left( \frac{R - r}{r} \cdot \theta \right)$$

$$y(\theta) = (R - r) \cdot \sin \theta - r \cdot \sin \left( \frac{R - r}{r} \cdot \theta \right)$$

Nákres a popis kružnic v kartézské souřadnici:

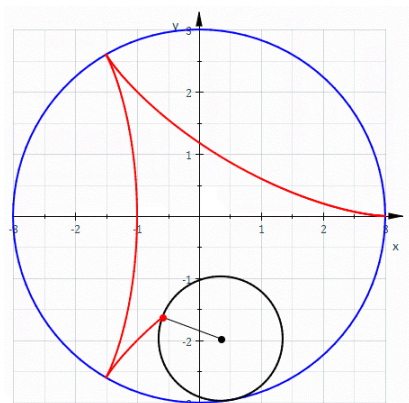


obr.3.7 hypotrochoida v kartézské souřadnici

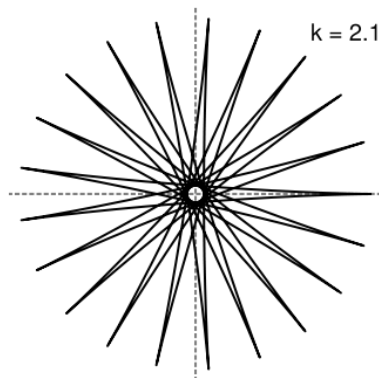
[9]



Příklady epicykloid:



obr.3.8 hypocykloida:  $R = 3$   $d = 1$   $k = 3$



obr.3.9 hypocykloida  $k = 2,1$  [8]

### 3.1.4 Hypotrochoida:

Je křivka, která vzniká pohybem bodu spojeného s kružnicí s poloměrem  $r$ , která se ovaluje podél vnitřního okraje statické kružnice  $R$ . Pohybující se bod je ve vzdálenosti  $d$  od středu obvykle menší kružnice s poloměrem  $r$ , která se ovaluje podél vnitřního okraje statické kružnice  $R$ . křivku lze vyjádřit v parametrickém tvaru jako:

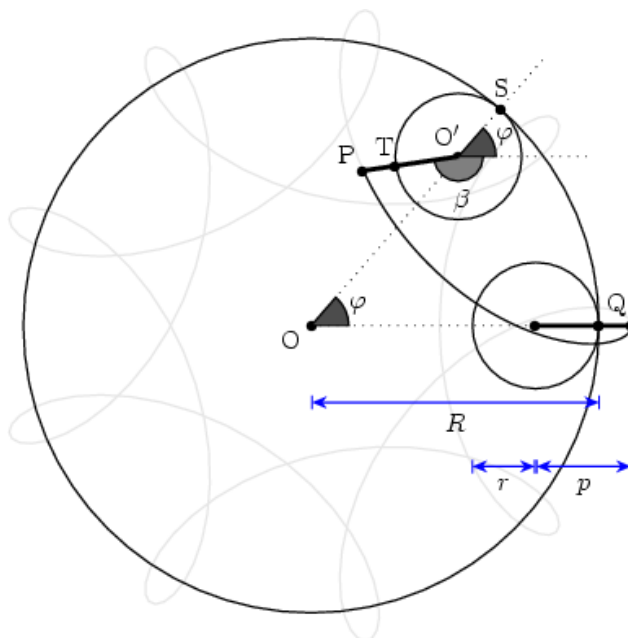
$$x(\theta) = (R - r) \cdot \cos \theta + d \cdot \cos \left( \frac{R - r}{r} \cdot \theta \right)$$

$$y(\theta) = (R - r) \cdot \sin \theta - d \cdot \sin \left( \frac{R - r}{r} \cdot \theta \right)$$

Kde  $\theta$  je úhel otáčení.

Pokud  $d = r$  (bod P se nachází přímo na vnitřní kružnici), nazývá se křivka hypocykloida.

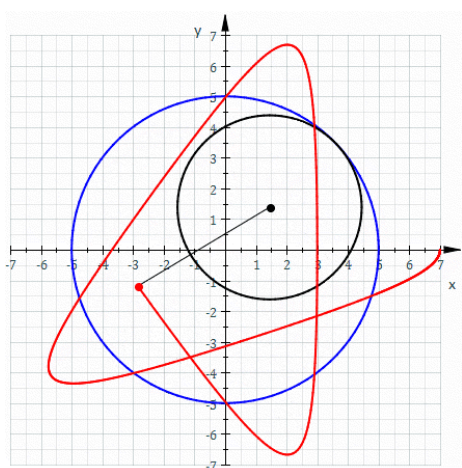
nákres a popis kružnic v kartézské souřadnici:



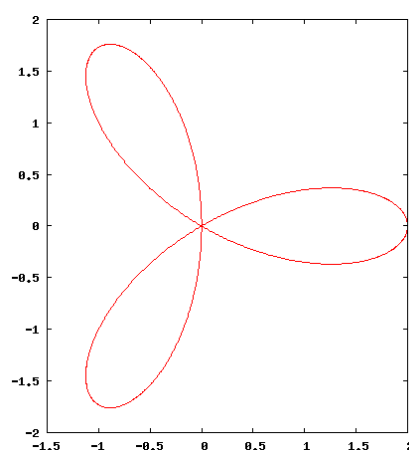
obr.3.10 hypotrochoida v kartézské souřadnici

[9]

Příklady hypotrochoid:



obr.3.11 hypotrochoida:  $R = 5$   $d = 4$   $k = 2,5$



obr.3.12  $R = 3$   $r = 1$   $d=2$

[8]

### 3.2 Spirograf

Obrazce, které jsou předmětem této práce se dají na papír vykreslit spirografem, který objevil britský inženýr Denys Fisher. Svůj vynález na vytváření těchto obrazců předvedl v roce 1965 na Norimberském národním trhu hraček. Spirograf poté začal ve své společnosti vyrábět. Distribuční práva byla získána firmou Kenner, Inc., která jej uvedla na americký trh v roce 1966.



obr.3. 14 šablona hračky spirograf

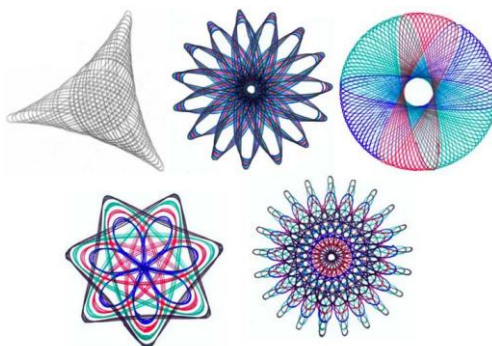
[11]

Provoz a princip šablony:

Spirograf se skládá z plastových převodů (ozubených koleček) různých velikostí, dále prstenů, trojúhelníků několika velikostí a s okraji jejichž ozubení zapadá do ostatních zoubků jiných částí a základní podložky. Například malá kolečka zapadají do vnitřních prstenů, ale také se můžou točit na vnější straně.

Při použití se pod plastovou základnu vloží papír a jedno z plastových koleček – známé jako stator, který je přitlačený na papír pod základnu. Další plastový kus – nazývaný rotor, je umístěný tak, že jeho zuby zapadnou do ozubení statoru. Množství kombinací spojení koleček navzájem je velké. Bod pera je umístěný v jednom z děr rotoru. Když se rotor točí a kopíruje svou hranou stator, pero vytyčí křivku.

[10]



obr.3. 14 obrazce, které lze vykreslit šablonou Spirograf

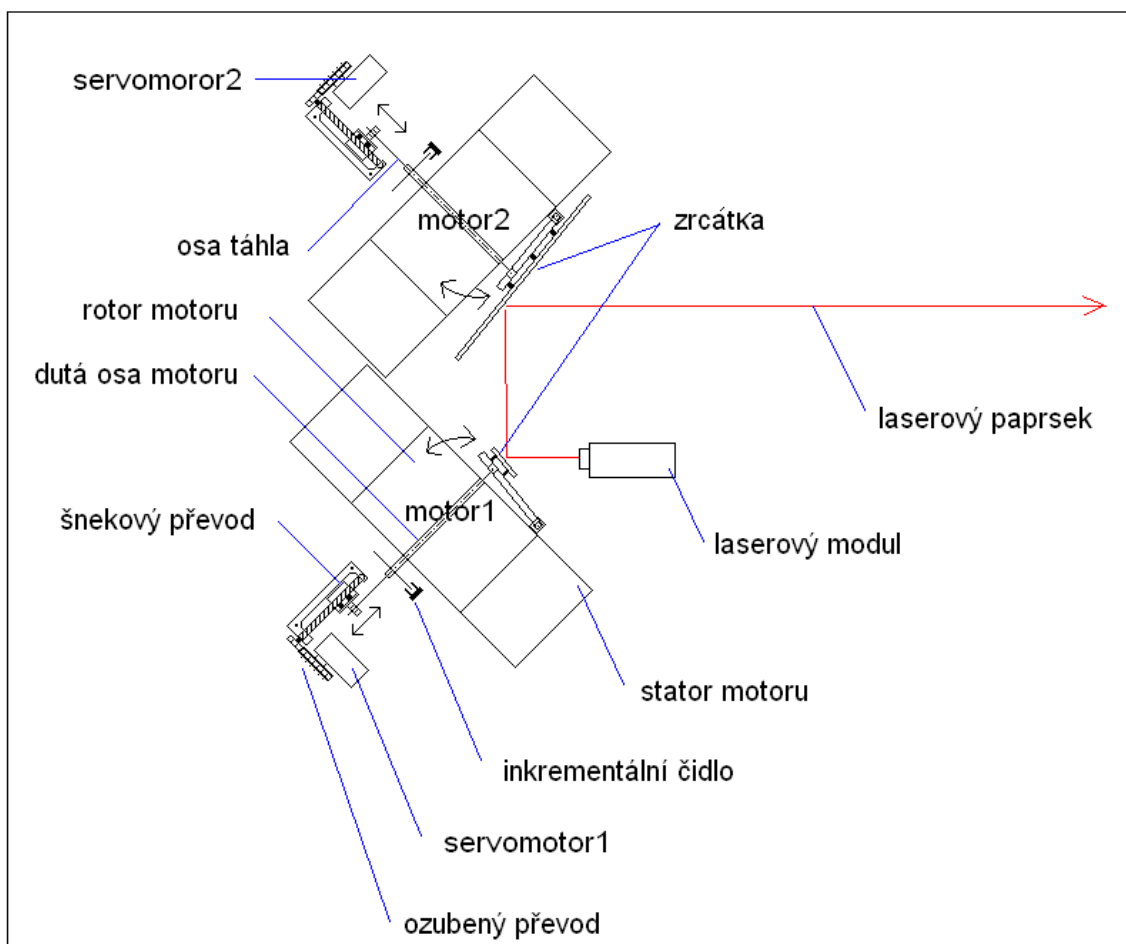
[10]

### 3.3 Návrh mechanického systému pro rozmítání laserového paprsku

#### 3.3.1 Hlavní princip a konstrukce

K tomu aby se na promítacím plátnu objevily, ať už statické nebo dynamické obrazce, je zapotřebí v našem případě rozpohybovat laserový paprsek a jeho dopadový bod natolik rychle aby setrvačnost oka zaznamenala souvislou křivku. Pro řešení tohoto problému byla zvolena metoda dvou rychle se točících zrcátek, protože tak poměrně snadno získáme rychle se pohybující odraz laserového svazku.

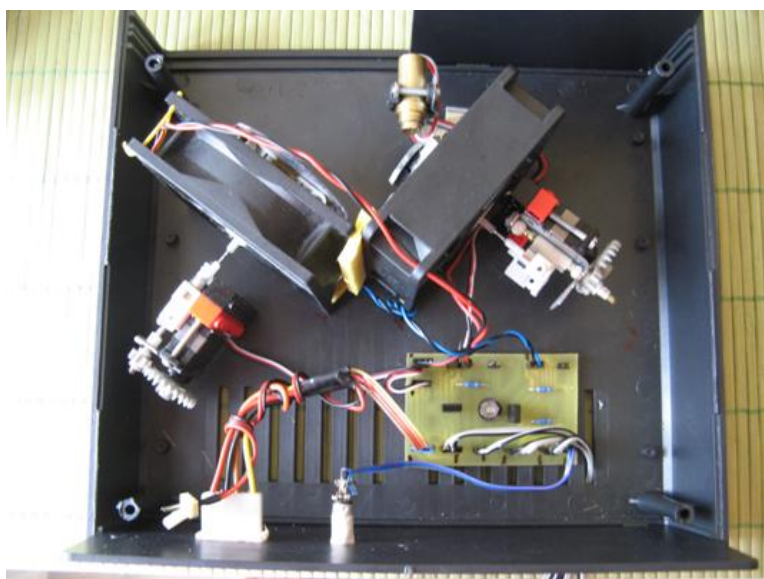
Zrcátka při pohybu odrážejí laserový paprsek tak, že se kopírují tvary kružnic geometrického systému např. hypotrochoidy. U zrcátek se mění zároveň úhel vychýlení od osy rotoru pomocí servomotorů, které tak dávají možnost vykreslit větší počet obrazců a plynule měnit jejich tvar. Z obr.3.15 a popisků je zřejmé rozmístění součástí rozmítacího zrcátkového systému.



obr.3. 15 náčrtek rozmítacího zrcátkového systému

### Konstrukce:

Jednotlivé části, jako jsou: laserový modul se stojanem, motory, deska plošného spoje jsou přišroubované do čtyřdílné plastové krabice typu KP 14, která má skleněný průzor. Rozměry krabice jsou: 235x217x92mm. Motory jsou uloženy na tlumící podložku, aby se nepřenášely velké vibrace, které jsou určitým zdrojem problémů. Úhel mezi motory není přesně 45°, protože jejich vzájemné natočení je sladěno tak, aby výsledný obraz byl co nejméně deformován a nebyl ořezáván okraji krabičky nebo laserovým modulem.



*obr.3. 16 fotografie odkryté vykreslovací jednotky*

#### 3.3.2 Použité motory:

Většina ventilátorů v dnešních počítačích používá velmi podobnou konstrukci a zapojení. Ventilátor se skládá ze základny, do které je vsazeno ložisko. To je nejčastěji kluzné (třecí) vyrobeno z bronzu nebo kuličkové. Na středový trn základny je nalisována deska s elektronikou a na ní je upevněna malá kostra se čtyřmi cívkami a kovovým jádrem.



*obr.3. 17 statory motorků klasických větráků do PC*

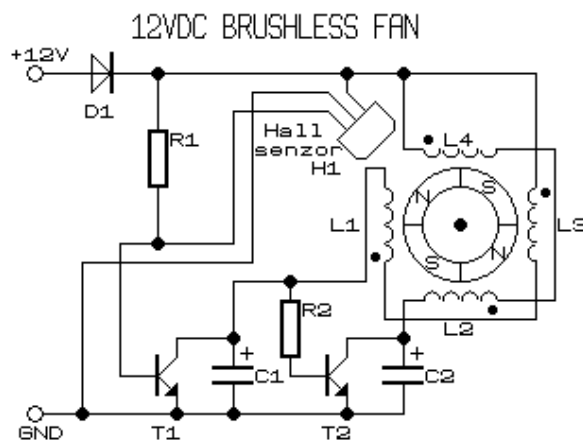
Menší ventilátory, např. na chlazení procesoru mají elektroniku s cívkami a ložiskem samonosnou a nalisovanou do umělohmotné základny. Cívky jsou speciálně vinuté a plechy složené, aby byly co nejtenčí. Elektronika je většinou v provedení SMD.

Vrtule má uprostřed ocelovou osu a na vnitřní straně výlisku je přilepen magnet ve tvaru prstence, který je na jednom místě přerušen a má 2 severní a 2 jižní póly. Osa má na konci vyfrézovaný zářez, do kterého se po nasunutí do ložiska z druhé strany nasune zajišťovací podložka, která zajistí vrtuli proti vysunutí. Tato podložka je nejčastěji vyrobena z plastu, nebo kovu.

#### elektronika statoru

Elektrické zapojení je ve všech větráčcích prakticky shodné a liší se jen v drobných detailech a hodnotách součástek. Základ tvoří Hallova sonda H1, která je citlivá na magnetické pole. Výstup sondy je přiveden do prvního tranzistoru T1, který spíná dvě protilehlé cívky L1 a L3. Na jeho výstup je také připojen přes odpor R2 druhý tranzistor T2 spínající zbývající dvě protilehlé cívky L2 a L4 umístěné kolmo na první dvě cívky. Na vstupu bývá také ochranná dioda D1, která zabráňuje nechtěnému přepólování napájení ventilátoru. Na výstup Hallovy sondy bývá připojen "pull-up" rezistor R1. Je to proto, že výstup je zapojen jako otevřený kolektor.

[4]



obr.3. 18 obecné schéma zapojení na desce statoru motoru

[4]

### popis funkce motoru větráku PC

Po zapnutí je vrtule v určité poloze. Můžou nastat dva případy: 1. Hallova sonda H1 není sepnutá, tranzistor T1 se díky R1 otevře a cívkami L1 a L3 prochází proud. T2 je uzavřen. Vrtule se dá do pohybu a až urazí maximálně čtvrt otáčky, tak se sepne Hallova sonda, T1 se zavře, T2 sepne cívku L2 a L4 a vrtule pokračuje v pohybu opět o čtvrt otáčky. Po té se Hallova sonda opět rozepne a celý postup se opakuje. Otáčky jsou přímo úměrné napájecímu napětí. Otáčky také závisí na hodnotách součástek, mechanickém tření, odporu vzduchu apod.

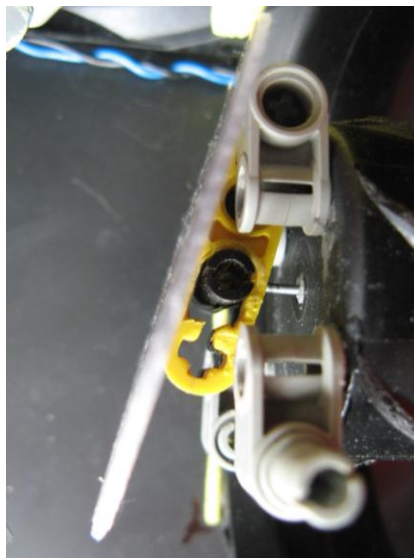
[4]

### úprava osy větráku

Aby se dal měnit úhel na rotačním zrcátku je vyměněna ocelová osa za plastovou dutou osu. V ose je táhlo, které servomotor společně s šnekovým převodem zasunuje a tak se mění úhel natočení na nosném rameni bez toho aby se významně omezovaly otáčky motorku. Motory jsou použity z ventilátorů pro elektroniku, protože u těchto typů motorů je lehce zaměnitelná osa za dutou, motor obsahuje kuličková ložiska a montáž ramene na rotor s magnetem také není složitá. Tyto větráčky dosahují rychlosti až k 2500 otáček za minutu, což by mělo stačit pro danou aplikaci.



3.19 rotor s dutou osou



obr.3.20 proměnné úhel natočení



### 3.3.3 Mechanická konstrukce posunovacího zařízení

Tato část má za úkol posunovat táhlo v obou směrech skrz dutou osu a měnit úhel, který svírá rameno, coby nosník zrcátka, s plochou rotoru. Rozsah pohybu táhla se pohybuje v milimetrech, proto je důležité přesné a jemné krokování. K regulaci posunu je použit nano-servomotor INO –LAB HGD-201HB. Na převod z točivého na posunovací pohyb je použito ozubení ze stavebnice lego a šnekový převod z CD-ROM mechaniky. Při plném mechanickém rozsahu (180°) samotného servomotoru je výchylka posunu 5mm. V samotném řízení pak bude rozsah o něco menší, důvodem je menší elektrický rozsah, který je menší než mechanický (kolem 170°). Po přepočtu na elektrický rozsah, by se měl změnit rozsah posunu na 4.72 mm.

Popis použitého nano-servomotoru:

Základní popis HGD-201HB:

- napájení : 4.8V(pouze)
- rychlost :  $0.07s \cdot 60^{\circ -1}$
- moment síly: 0.9 kg.cm
- velikost : 23.0x11.4x23.7 mm
- hmotnost : 9.83 g

INO –LAB HGD-201HB



obr.3. 21 použitý servomotor [12]

Uplatnění těchto motorků je především v aplikacích, kde vyžadujeme takřka okamžité naklonění hřídele (RC modely vrtulníků, letadel, aut atd.). Rychlost je zde důležitá a udává se ve vteřinách, kdy se rameno otočí o 60°. Tyto časy jsou v řádech desítek milisekund. Takovéto motorky již v sobě obsahují mikrokontrolér, který nastavuje úhel natočení podle délky signálu přicházející na PWM vstup. Řádově se tyto časy šířky impulsů pohybují od 1ms do 2ms, kde každý s těchto extrémů vyhovuje maximální výchylce. Dále můžeme výchylku řídit střídou signálu, která se dá odvodit z pulsu v log 1 a frekvence PWM:

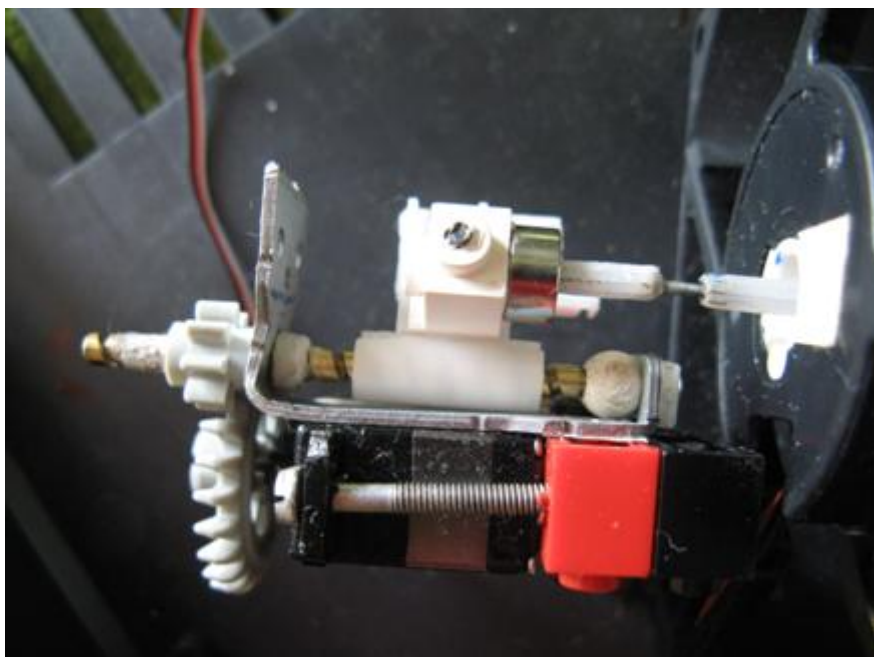
$$St = \frac{t_{MAX}}{t_{HIGH} + t_{LOW}} \cdot 100\% = t_{MAX} \cdot \frac{1}{T} \cdot 100\% = t_{MAX} \cdot f \cdot 100\%$$

Mezera mezi impulsy s  $t_{HIGH}$  je tolerována, v daných mezích 8-22ms, protože jde o digitální servomotor, u analogových je mezera vcelku nepodstatná vůči přesnosti nebo nefunkčnosti. u těchto servomotorů je standardní frekvence PWM 50Hz. Na trhu jsou dostupná analogová a digitální, samozřejmě jsou digitální přesnější, ale také mají vyšší odběr proudu.



Krouticí moment, síla a rychlost je také dimenzována daným typem převodovky, která u těchto malých typů (11mm) bývá až 0.9kg.cm. I tyto malé motory obsahují kuličková ložiska, karbonové ozubení v převodovce, hliníkový chladič a odolný potenciometr. Motorek má tedy 2 vodiče pro DC napájení a jeden signální vodič pro řízení.

U zmíněného servomotoru bylo zjištěno, že při nastavování výchylky ramene pomocí PWM s krajními hodnotami šířky pulsů udávanými výrobcí (1ms , 2ms) je mechanický rozsah využít jen z tří čtvrtin. Při testování jsem dosáhnul téměř plného mechanického rozsahu (necelých 180°) při hodnotách od 0.750ms a 2,3ms. Motorek však při testování zvládal frekvence PWM i 153 Hz (standardně je 50Hz), to byla nejmenší frekvence kterou bylo možné na DSP příslušného typu nastavit. Mezera mezi pulsy byla nejmenší 4.2ms, což neodpovídá kritériu výrobců. Při zapojení na napájecí napětí se rameno zablokuje i bez řídicího signálu, ale síla kterou dokáže rameno servomotoru vyprodukovat je řádově nižší než síla, daná chtěnou výchylkou (přívodem PWM signálu).



obr.3. 22 *Pohled na převody šnekového posunovacího zařízení poháněné servomotorem*

### **3.4 Laserový modul**

Laserové diody (LD) jsou zdroje optického záření s kvalitativně novými vlastnostmi ve srovnání s nekoherentními zdroji. Optické záření generované laserem soustředěné do velmi úzkého intervalu vlnových délek, je do značné míry koherentní a samotný laser se vyznačuje vysokou zářivostí a malou rozbíhavostí laserového svazku.

#### Funkce laserových diod

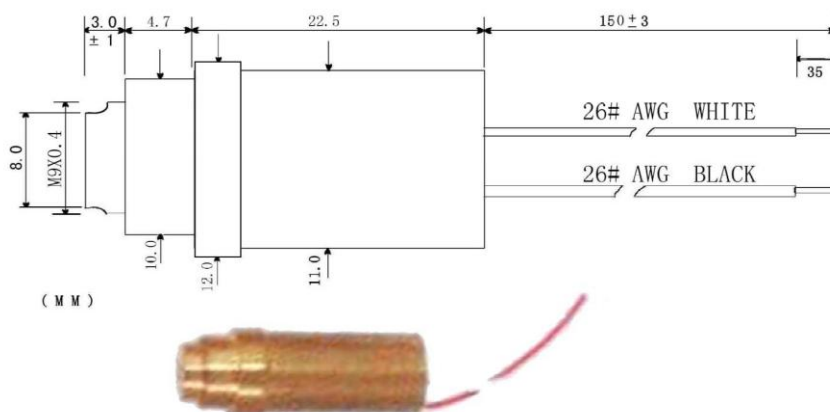
Základem funkce všech laserů je proces stimulované emise. U polovodičových laserů je však tento proces specifický v tom, že v polovodičových krystalech jsou aktivní atomy hustě vedle sebe a zářivé přechody se neuskutečňují mezi diskretními energetickými hladinami, ale mezi energetickými pásy. Elektrony obsazují nejnížší energetické hladiny ve vodivostním pásu až po Fermiho kvazihladinu pro elektrony  $E_{fn}$ . Stavby ve valenčním pásu jsou bez elektronů až po Fermiho kvazihladinu pro díry  $E_{fp}$  (plyne z principu zachování neutrality náboje). Při dopadu fotonů na polovodič s energií větší než  $E_g$  (šířka zakázaného pásu (eV)), ale menší než  $E_{fn} - E_{fp}$  nemůže dojít k jejich absorpci, protože hladiny, na něž by se mohl uskutečnit přechod spojený s pohlcením fotonu, jsou již obsazeny. Dopadající fotony tedy mohou stimulovat přechody elektronů z vodivostního do valenčního pásu s následnou emisí fotonů nerozlišitelných od budících.

Aktivní prostředí v „injekčních“, polovodičových laserech vzniká při injekci elektronů a děr z přechodu PN nebo z heteropřechodu. Optické záření je pak generováno, jak je tomu i ve všech generátorech, v důsledku zavedení kladné zpětné vazby, která část zesíleného signálu z výstupu přivádí na vstup. K tomu se používají různé typy rezonátorů (např. Fabry-Perot), selektivních odrazečů nebo rozložené zpětné vazby.

Při nízkých proudech tekoucích přes přechod PN v přímém směru je generováno spontánní záření šířící se ve všech směrech s náhodnou fází. Růst proudu zvyšuje rychlost zářivé rekombinace; to vede k růstu hustoty fotonového toku. Generované fotony stimulují další rekombinace. Vzhledem k tomu, že největší počet generovaných fotonů má energii rovnou energii maxima spektrálního rozdělení spontánní emise, nastává pro tuto energii nejvíc vynucených přechodů ve srovnání s jinými oblastmi spektrálního rozdělení. Tato okolnost vede k postupnému zužování spektra spontánní emise a k výraznému růstu intenzity vyzařování v oblasti maxima spektrálního rozdělení emise. Roste-li intenzita vyzařování nelineárně s růstem buzení, nazývá se tento proces superluminiscence. Fotony generované v procesu superluminiscence se šíří stejně jako ve spontánním režimu ve všech směrech s náhodnou fází. Přechod k laserovému režimu nastává, když se stimulované zesílení rovná ztrátám a záření se stává koherentním. Koherence se dosahuje použitím optického rezonátoru, který zajistí selektivní zesílení elektromagnetické vlny s určitou frekvencí a definovanou fází, čímž vzniká stojaté vlnění. Stupeň koherence je dán typem a kvalitou použitého rezonátoru.

[6]

Použitý laserový modul F-LASER 5MW Obsahuje laserovou diodu 650nm s optikou a stabilizátorem proudu.



obr.3.23 rozměry laserového modulu

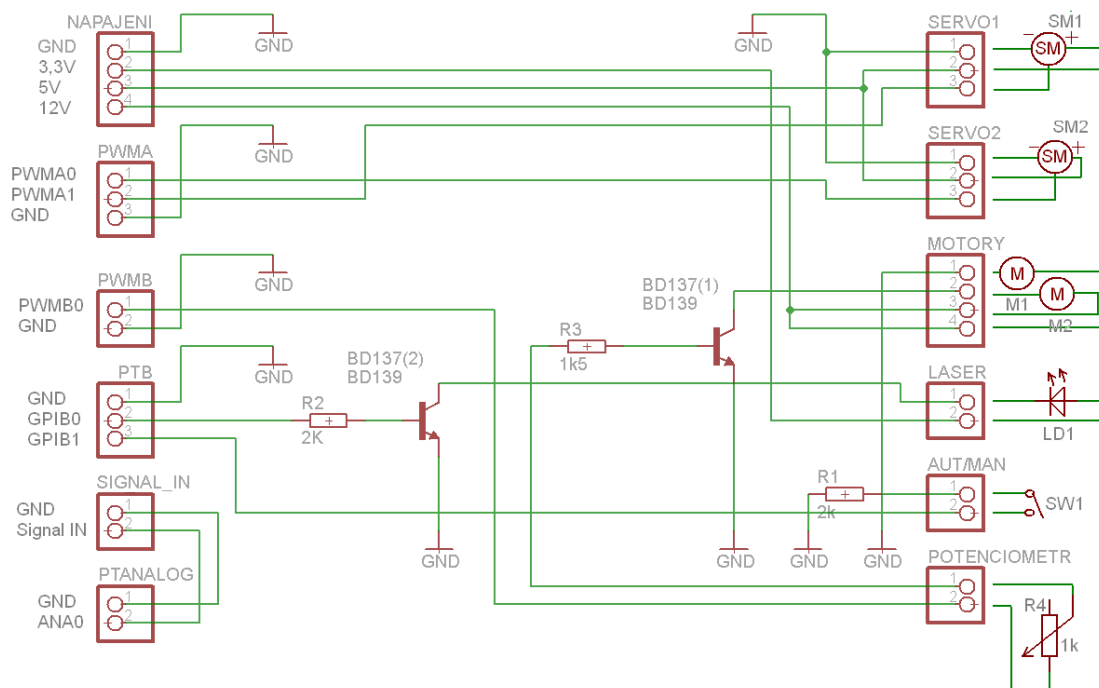
Tabulka vlastností laserového modulu F-LASER 5MW:

PARAMETRY	MIN	TYPIA	MAX	UNIT
Optický výkon	2	3	4	mW
Vstupní napětí	---	3.0	---	V
Odebíraný proud	---	25	35	mA
Vlnová délka paprsku	---	650	---	nm
Provozní teplota	-10	---	40	°C
Skladovací teplota	-20	---	60	°C
Maximální rozptyl svazku na 6 metrů	---	6x10	13x15	mm
životnost	3000	---	---	hodin

[5]

### 3.5 Spínací elektronika výkonových částí

Je zcela zřejmé, že výstupní obvody DSP řídící motory, servomotory a laserový modul, nebudou dostatečně výkonné na to aby byly schopny zásobovat proudem všechny tyto členy vykreslovací jednotky. Laserový modul a motor2 je spínán tranzistorem BD137. Deska plošného spoje je osazena také napájecím konektorem. Ten má 3 vstupní napětí: 3.3V pro laser, 5V pro servomotory, 12V pro motory. Jako zdroj byl proto použit 300W zdroj z PC. Je zde také potenciometr, kterým lze doladit otáčky motoru2, přepínač režimů (automat / manuál).



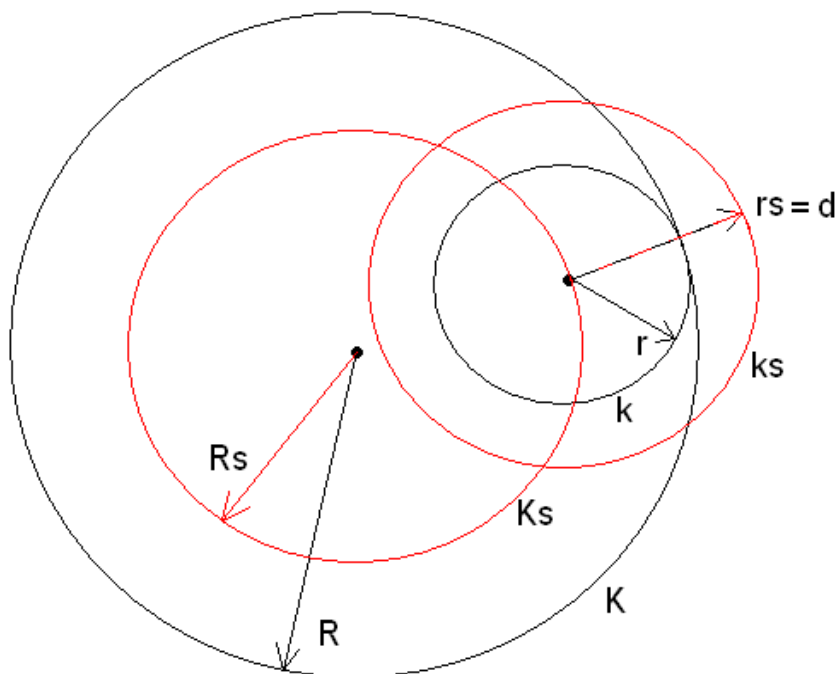
obr.3. 24 schéma zapojení elektroniky v rozmitacím modulu

#### **4. Přepočet parametrů z geometrického systému na veličiny vykreslovací jednotky**

Z teorie vykreslení hypotrochoidy pomocí dvou ovalujících se kružnic vyplývá, že tato metoda lze s určitými rozdíly přenést do mechanické podoby a vykreslit tak obrazec laserovým paprskem. Rozdíl ve vytvořeném rozmitacím systému je ten, že statická vnější kružnice nevytváří obvod podél níž se ovaluje vnitřní kružnice, ale vytváří kružnici, na které se pohybuje střed kružnice vnitřní.

Pokud si chceme vykreslit hypotrochoidu s parametry jaké jsou popsány v teorii, tedy poloměry kružnic, kde se vnitřní kružnice  $k$  s poloměrem  $r$  ovaluje vně statické kružnice  $K$  s poloměrem  $R$  a vzdálenost vykreslovacího bodu od středu ovalující se kružnice je vzdálenost  $d$ , je nutné přepočítat parametry.

#### 4.1 Převod parametrů obrazců z geometrického na zrcátkový systém



obr.4. 1 kružnice drah vykreslování pro geometrický i zrcátkový systém

Z nákresu (černé kružnice - geometrický systém, červené kružnice - zrcátkový systém) lze vyvodit vztahy:

$$Rs = R - r$$

$$rs = d$$

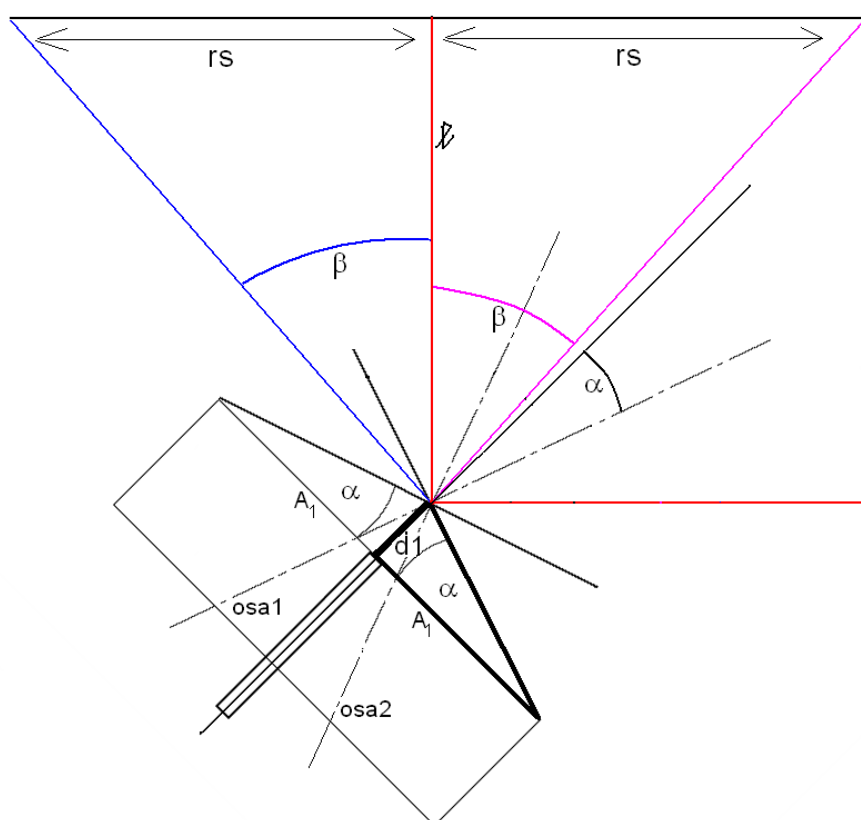
Pokud roztočíme zrcátko s osou odchýlenou od ose otáčení na něhož svítíme laserovým paprskem, vytvoří se na dopadovou plochu kružnice  $ks$ . Tuto kružnici aproximujeme jako vzdálenost vykreslujícího bodu od kružnice  $d$ . Pokud-li bude dopadová plocha druhé rotující zrcátko s osou odchýlenou od ose otáčení, rotující opačným směrem než je směr vykreslování první kružnice, vytvoří druhé zrcátko pomyslnou kružnici  $Ks$ , po které se pohybuje střed kružnice  $k$  geometrického systému hypotrochoidy.

## 4.2 Přepočítání parametrů obrazců na říditelné veličiny motorů a servomotorů

V této části si odvodíme velikost strany  $d$ , trojúhelníku ramene zrcátka, kterou můžeme měnit pomocí servomotoru. Odvodíme také otáčky motorů.

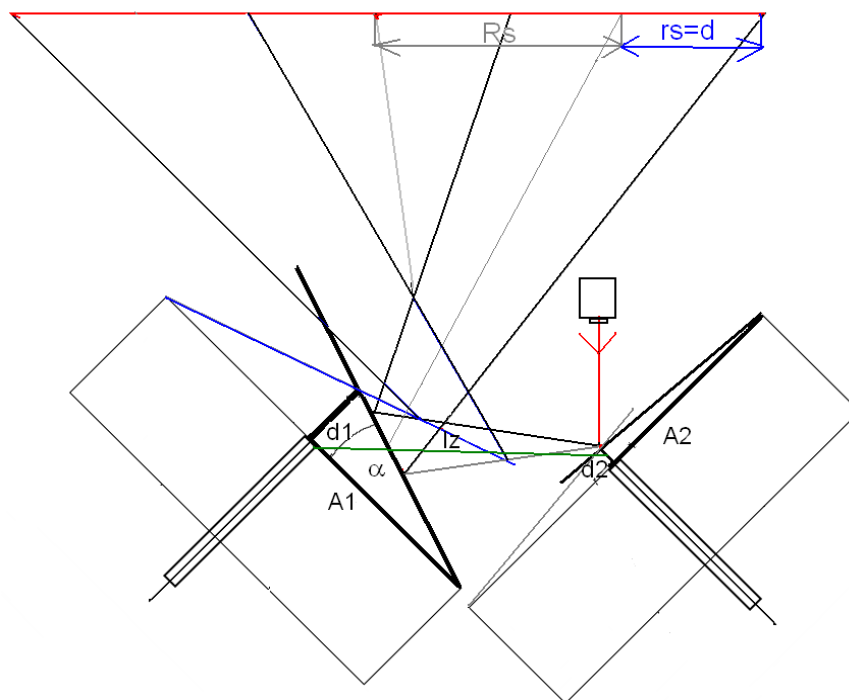
Ze schéma tu je zřejmé, že při změně úhlu v trojúhelníku ramene se mění lineárně úhel v pomyslném trojúhelníku vytvářeného paprskem laseru. Z pravidla které hovoří o tom že úhel dopadu se rovná úhlu odrazu jsme schopni odvodit že úhel  $\beta$  je dvojnásobkem úhlu  $\alpha$ . Tedy:

$$\beta = 2 \cdot \alpha$$



obr.4. 2 náčrtek odrazů paprsku laseru při točivém pohybu rotoru 1

V náčrsku odrazů paprsku laseru při točivém pohybu rotoru 1 je goniometricky znázorněno jak se vykresluje kružnice  $k$  s poloměrem  $r$  v zrcátkovém systému. Červeně je označen paprsek laseru kde  $d = 0$ ,  $\alpha = 0^\circ$ , v tomto případě je osou úhlu mezi dopadem a odrazem osa rotoru. Vychýlí-li se rameno zrcátka o úhel  $\alpha$  i o tento úhel se vychýlí také osa dopadu a odrazu laserového svazku a stane se jí  $osa1$ .



obr.4. 3 náčrtes odrazů laseru přes obě zrcátka

1. Vyjádříme  $d_1$ :

- Vyjádření  $\alpha$  z trojúhelníku paprsků:

$$\tan \beta = \frac{rs}{l+l_z} \rightarrow 2 \cdot \tan \alpha = \frac{rs}{l+l_z} \rightarrow \tan \alpha = \frac{rs}{2 \cdot l+l_z}$$

- Vyjádření  $\alpha$  z trojúhelníku ramene rotoru:

$$\tan \alpha = \frac{d_1}{A_1} \quad \begin{array}{l} Rs = R - r \\ rs = d \end{array}$$

- Dosazení  $\alpha$  z trojúhelníku paprsků do trojúhelníku ramene rotoru a vyjádření  $d_1$ :

$$\frac{rs}{2 \cdot (l+l_z)} = \frac{d_1}{A_1} \rightarrow d_1 = \frac{d \cdot A_1}{2 \cdot (l+l_z)}$$

## 2. Vyjádříme $d_2$ :

- Vyjádření  $\alpha$  z trojúhelníku paprsků:

$$\tan \beta = \frac{Rs}{l} \rightarrow 2 \cdot \tan \alpha = \frac{Rs}{l} \rightarrow \tan \alpha = \frac{Rs}{2 \cdot l}$$

- Vyjádření  $\alpha$  z trojúhelníku ramene rotoru:

$$\tan \alpha = \frac{d_2}{A_2} \qquad \begin{array}{l} Rs = R - r \\ rs = d \end{array}$$

- Dosazení  $\alpha$  z trojúhelníku paprsků do trojúhelníku ramene rotoru a vyjádření  $d_1$ :

$$\frac{R-r}{2 \cdot l} = \frac{d_2}{A_2} \rightarrow d_2 = \frac{(R-r) \cdot A_2}{2 \cdot l}$$

## 3. Vyjádříme otáčky motorů:

Protože v DSP se neplátuje programovat regulátor, který by udržoval předepsané otáčky, stačí experimentem zjistit střidu spínání motoru2, při které se vytváří obrazce, které odpovídají svým tvarem poměru mezi  $R$  a  $r$ . Motor1 je připojen na napětí 12V, dosahuje tak maximálních otáček.

Experimenty bylo dále zjištěno, že při změně velikosti  $d$  u motoru 2 dochází k omezování otáček v důsledku překonávání odporu vzduchu plošně velkým zrcátkem a jeho hmotností, která byla již omezena náhradou zrcátka za reflexní fólii nalepenou na úzký plast kruhového tvaru. Byla vytvořena tabulka závislosti střidy na velikosti  $d$  pro různé poměry poloměrů kružnice  $K$  geometrického systému. Její podoba je přenesena do programu v C# jako pole, kde se odvozuje střída od tabulkových hodnot a posílá do DSP jako jeden z parametrů.

Otáčky motoru 1 se při změně  $d$  mění jen nepatrně a proto se tento rozdíl otáček u tohoto systému zanedbává. Důvodem je malé zrcátko s malým odporem vzduchu.



## **5. Základní problematika DSP**

### **5.1 Úvod**

Digitální signální procesor nebo také Digitální signálový procesor (zkratka DSP) je mikroprocesor, jehož návrh je optimalizován pro algoritmy používané při zpracování digitálně reprezentovaných signálů. Hlavním nárokem na systém bývá průběžné zpracování velkého množství dat "protékajících" procesorem.

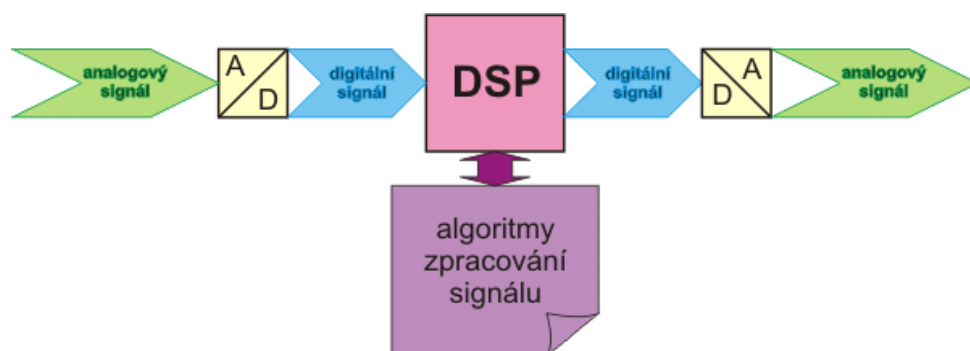
Digitální signálové procesory již nacházejí uplatnění v procesech, v nichž je třeba měřit, analyzovat, vyhodnocovat, uchovávat či jen předzpracovat značné množství informací a v určité fázi zpracování vyvodit akční zásah. Vysoký výpočetní výkon tyto procesory předurčuje i pro velmi náročné aplikace, které vyžadují řízení v reálném čase. Tyto procesory jsou specializovány na rychlé provádění aritmetického násobení, sčítání, rotací a přesunových instrukcí, což jsou operace, na nichž je založeno číslicové zpracování signálů. Tomu je přizpůsobena struktura i instrukční soubor DSP. Prošly několika vývojovými etapami a univerzální i specializované signálové procesory nacházejí stále širší pole použití v komunikačních digitálních systémech, digitálním záznamu zvuku, zpracování řeči a signálů v lékařské elektronice, televizi, navigaci a radiolokaci. Jen pro příklad, v přijímači digitálního rozhlasu je použito šest DSP. V nemalé míře se signálové procesory také využívají v aplikacích průmyslové elektroniky. Příkladem mohou být řídicí systémy výkonových měničů a elektrických pohonů.

[2]

### **5.2 Obecná aplikace**

Jedním ze základních důvodů k vytvoření DSP byla ta skutečnost, že klasické analogové obvody sloužící pro zpracování signálu bývají náročné z hlediska návrhu, nastavení, provedení a reprodukovatelnosti, přičemž jakmile jsou vyrobeny nelze jejich funkci téměř modifikovat. Ve srovnání s tím cena "digitálního" křemíku stále klesá a výkon roste.

Na obrázku je typické blokové schéma zařízení využívajícího DSP. Analogový signál je nejprve převeden A/D převodníkem na digitální a v této podobě je průběžně zpracováván digitálním signálním procesorem. Zpracovaný digitální signál je D/A převodníkem zpět převeden na analogový. V mnoha zařízeních prochází signál tímto řetězcem v reálném čase, ale na některých signálech je potřeba provést tak složité a výpočetně náročné algoritmy, že to ani velmi rychlý DSP procesor v reálném čase nestihne a digitalizovaná data musí být nejprve zaznamenána do paměti a odtamtud teprve postupně zpracovávána.

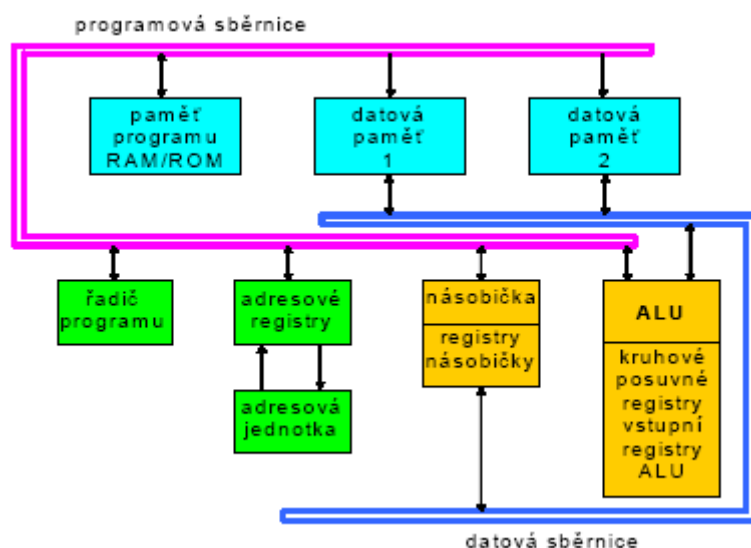


obr.5. 1 Typický řetězec pro zpracování signálu v DSP

[2]

### 5.3 Architektura

Signálový procesor DSP má obdobné vlastnosti a způsob programování jako univerzální procesor, ale jeho architektura a instrukční soubor právě podporují rychlé a účinné provedení matematických operací, které se uplatňují ve zpracování signálů. Mezi tyto operace patří spektrální analýza provedená algoritmem rychlé Fourierovy transformace (FFT), komprese obrazu transformací wavelet, práce s polynomy, maticemi apod. Většina DSP má harvardskou architekturu s odděleným zpracováním instrukcí a dat. Toto uspořádání dovoluje současné provádění operací s daty a adresami. DSP má redukovanou instrukční množinu a je vybaven větším počtem výkonných jednotek, pamětí a vnitřních sběrnic.



Obr.5.2 modifikovaná harvardská architektura

Vysoká rychlost výpočtu se dosahuje obdobnými cestami jako u univerzálních mikroprocesorů, a to zvyšováním hodinového kmitočtu, architekturou procesoru a paralelním zpracováním úloh více procesory. Obecně dokáží DSP provést nejméně jednu operaci násobení a sčítání v jednom instrukčním cyklu. U mnoha DSP je navíc možné násobičku (např.  $16 \times 16$  b  $\uparrow$  32 b) fiktivně rozdělit např. na dvě jakoby nezávislé ( $8 \times 8$  b  $\uparrow$  16 b) a jednou instrukcí provést dvě operace násobení s menší „přesností“. Obdobně lze rozdělit aritmeticko-logickou jednotku. V závislosti na architektuře dokáží DSP ještě v tomtéž instrukčním cyklu provést např. operaci nepřímého adresování, bitových posuvů či rotací, vyhodnotit podmínku vykonání instrukce, příp. jiné operace. Jak už bylo řečeno, mají signálové procesory modifikovanou harvardskou architekturu s minimálně třemi paralelně pracujícími aritmetickými jednotkami. Jedna slouží ke zpracování dat, druhá k výpočtu adres a třetí umístěná v řadiči k obsluze programového čítače a k obvodové podpoře programových cyklů. Modifikace je dále podpořena několikanásobným sběrnicevým systémem. Dalšími možnostmi pro zvýšení výpočetního výkonu signálových procesorů je využití architektury VLIW (Very Long Instruction Word), superskalární architektury a paralelních systémů. Pro omezení častých přístupů do relativně pomalých externích pamětí bývá někdy integrována instrukční paměť cache (i dvoustupňová), případně i datová paměť cache. Ve srovnání s procesory pro všeobecné použití mají DSP menší spotřebu, nižší cenu a vyšší výkon v úlohách zpracování signálů. Většina instrukcí trvá jeden instrukční cyklus a DSP má garantovanou dobu, během níž se dostane do smyčky aktivního přerušení (např. čtyři instrukční cykly), což je přínosem pro systémy reálného času. Na druhou stranu se u DSP s rostoucím výpočetním výkonem rozrůstá jejich architektura o další výpočetní jednotky, další sběrnice a zvětšuje se hloubka zřetězení (pipelining), což přinejmenším snižuje efektivitu využití procesoru. Největšími výrobci jsou firmy Motorola, Texas Instruments, Analog Devices a další. Všechny pak mají v nabídkách jak samotné signálové procesory, tak i vývojové kity s těmito čipy a vývojová prostředí pro ladění a implementaci algoritmů. Samotné procesory mohou mít na čipu integrované různé typy paměti RAM, ROM a FLASH.

[2]

#### **5.4 Periférie**

Digitální signálový procesor obdobně jako jiné mikroprocesory obsahuje periférie I/O, pomocí kterých je možná komunikace procesoru s vnějším prostředím. Tento procesor, samozřejmě vzhledem k druhu aplikací, na které je uzpůsoben, obsahuje odlišné periférie než běžné procesory. K těmto procesorům je možné také připojení externích pamětí FLASH, SDRAM, díky externí sběrnicevým periférii, dále je možné připojit externí krystal řídicí frekvenci procesoru, ale u mnoha DSP je implementován i vnitřní krystal. Samozřejmostí je také adresová, datová sběrnice a DMA kanály, které slouží k přenosu dat, instrukcí, adres v procesoru. Ostatní periférie jsou přímo určeny ke komunikaci s vnějším prostředím, sběru a generování dat a signálu:

- JTAG port – speciální periférie pro řízení programu v procesoru, jeho ladění a nahrávání do paměti, tedy komunikace mezi vývojovým osobním počítačem a programovaným DSP.
- Sériové periférie – určené k sériové komunikaci. Je možné setkat se s několika typy periférií, odlišných v komunikačním protokolu SPI, SCI, PCI, UART
- Průmyslové periférie – určené ke komunikaci v průmyslových aplikacích, kde může dojít k rušení okolním agresivním prostředím např. CAN.
- Digitální vstupy a výstupy (GPIO, DIO)
- AD/DA převodník
- PWM moduly, časovače/čítače

Digitální signálové procesory DSP jsou v mnoha případech signálového řízení nesporně výhodnější, rychlejší a levnější než jakékoliv jiné hardwarové řešení (FPGA, konvenční mikroprocesor). Oproti standardním mikroprocesorům umožňuje DSP komplikovanější matematické výpočty a dokáže zpracovávat signál o vyšší frekvenci oproti běžným procesorům. Ve srovnání s procesory běžných i průmyslových osobních počítačů je DSP méně náročný na příkon, tedy na chlazení a spotřebu, ale vzhledem k rychlosti a složitosti výpočtů je srovnatelný. Při výrobě moderních DSP je používána CMOS technologie, zaručující příkon řídicího systému dokonce pod 1W. Srovná-li se použití DSP a zapojení klasických analogových obvodů, je v mnoha aplikacích výhodnější DSP pro svou vysokou přesnost, flexibilitu, schopnost adaptace pomocí softwaru, kde není většinou nezbytné měnit elektronické komponenty. U moderních DSP je možné řídit signály s maximální frekvencí až několika MHz.

[3]

## **5.5 Programování**

Psaní kódu pro DSP je dnes běžné jazykem symbolických adres a jazykem C. Výhodou jazyka symbolických adres je, že výsledný kód bude nejspíše kratší a procesor jej rychleji provede. Obtížně se však pomocí něj programují velké projekty, na kterých může pracovat i více programátorů. Naopak v jazyce C bude velký projekt rychleji napsaný, ale bude na DSP

probíhat pravděpodobně pomaleji. Jazyk C je vhodný pro DSP se složitější architekturou (VLIW, CLIW a superskalární) a jeho výhodou je snadnější přenositelnost kódu na jiný DSP, ale také jsou zde vytvořeny speciální knihovny pro DSP funkce, které velmi usnadňují a urychlují programování. V poslední době se objevují nástroje pro grafický vývoj aplikací, kde jsou jednotlivé části programu tvořeny bloky propojenými šipkami naznačujícími návaznost programu a každý blok má funkci definovanou jazykem C, příp. jazykem symbolických adres.

[2]

## **5.6 Rozdělení**

Základním dělením digitálních signálních procesorů je dělení podle použité aritmetiky. Existují DSP pracující:

- v celočíselné aritmetice
- v aritmetice s pevnou řádovou čárkou
- v aritmetice s plovoucí řádovou čárkou

Procesory s celočíselnou aritmetikou jsou sice levné, ale algoritmy výpočtů stále narážejí na nutnost převádět reálná čísla na celá a mezivýsledky výpočtů se musí neustále upravovat tzv. normalizacemi. Proto je vývoj algoritmů v těchto typech procesorů výrazně náročnější. Hodí se proto zejména pro masovou produkci výrobků, kde nevadí poněkud vyšší cena vývoje, ale důležitá je zejména cena samotné součástky.

Procesory s plovoucí řádovou čárkou jsou sice složitější a dražší, ale vývoj softwaru je pro ně výrazně jednodušší. Nevýhodou zde může být rovněž vyšší spotřeba energie.

Procesory s pevnou řádovou čárkou mohou sice být určitým kompromisem, ale prakticky je nelze jasně odlišit od procesorů pracujících v celočíselné aritmetice.

Dalším kritériem pro dělení digitálních signálních procesorů je šířka jejich datové sběrnice. Ta bývá od 16 bitů výše. Další dělení může být na jedno jádrové nebo více jádrové DSP.

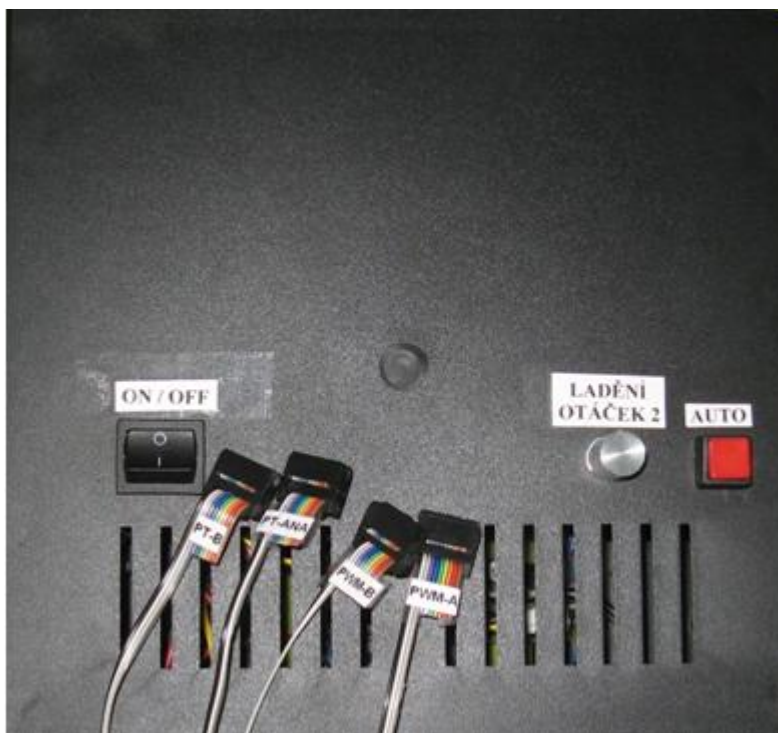
[2]

## **6. Hardware řídicího systému pro vykreslovací jednotku**

### **6.1 Blokový popis systému**

V konečném návrhu systému pro vykreslovací jednotku, používající laserový svazek paprsku a zrcátka k zobrazení statických i dynamických obrazců, jsou dohromady integrovány tyto části hardwaru:

1. Rozmítací systém ( zrcátková soustava vykresluje laserem obrazce)
  - Tento systém je schopný vykreslovat určitý typ geometrických obrazců s libovolnými parametry.
  - Obsahuje dva motory na 12V, z toho jeden je připojen přímo na Ucc a druhý motor je řízený DSP a má regulované otáčky pomocí PWM (10 KHz).
  - Součástí jsou také dva servo-motory na 4,8V, které řídí DSP pomocí PWM (53Hz) a šířky pulsů v log.1.
  - Obsahem je hlavní vypínač od el. proudu a přepínač, kterým lze měnit mezi automatickým a manuálním režimem. Ten souvisí s připojením k PC.
  - Potenciometrem lze doladit předepsané otáčky motoru2
  - Přítomen je laserový modul. Ten je aktivován při správném běhu DSP a je jím také ovládán.
  - Všechny výkonové výstupy jsou spínány tranzistory BD137 na desce plošného spoje, která má i konektory na propojení se vstupy a výstupy programovacího kitu a je součástí rozmítacího systému.



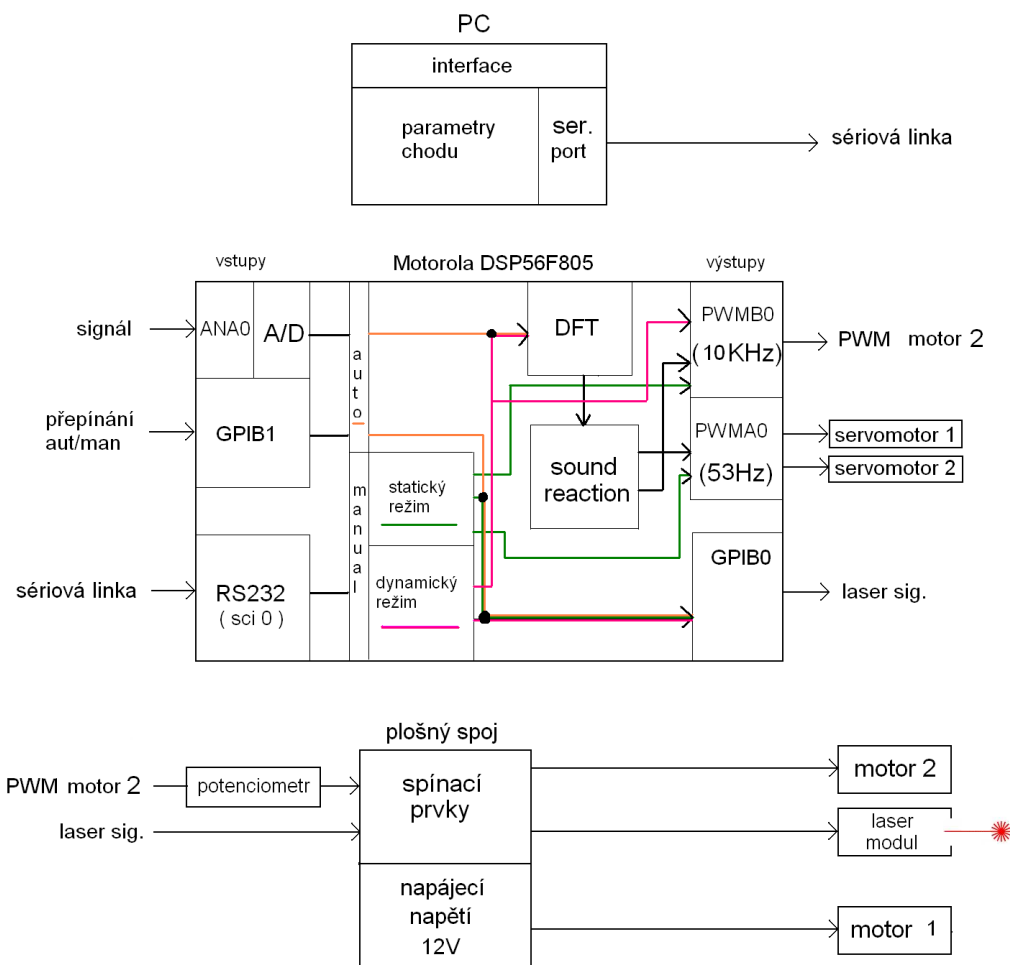
obr.6.1 rozmětacího systém - pohled na funkční ovládací prvky a konektory

## 2. DSP ( zpracování signálů, řízení periférií a výstupů )

- Byl vybrán programovací kit značky Freescale semiconductor s DSP typu DSP56F805.
- Vzorkuje signál ze kterého počítá pomocí DFT (je interní funkcí) frekvenční spektru. Podle jeho tvaru řídí tyto periferie: digitální výstupy na portu B, PWM výstupy na portech PWMA a PWMB.
- Přijímá řídicí data z PC přes rozhraní RS232.

## 3. PC ( jednosměrná komunikace z PC do DSP )

- Počítač mimo zasílání dat s parametry chodu přepočítává souřadnice z geometrického systému na systém, který odpovídá navrženému rozmítacímu systému ve statickém režimu.
- Je možnost použít výstup zvukové karty k přívodu signálu do DSP.



obr.6.2 blokové schéma vstupů ,výstupů, periférií a jednotlivých bloku v programu:

## 6.2 Technické parametry vývojového kitu DSP56F805

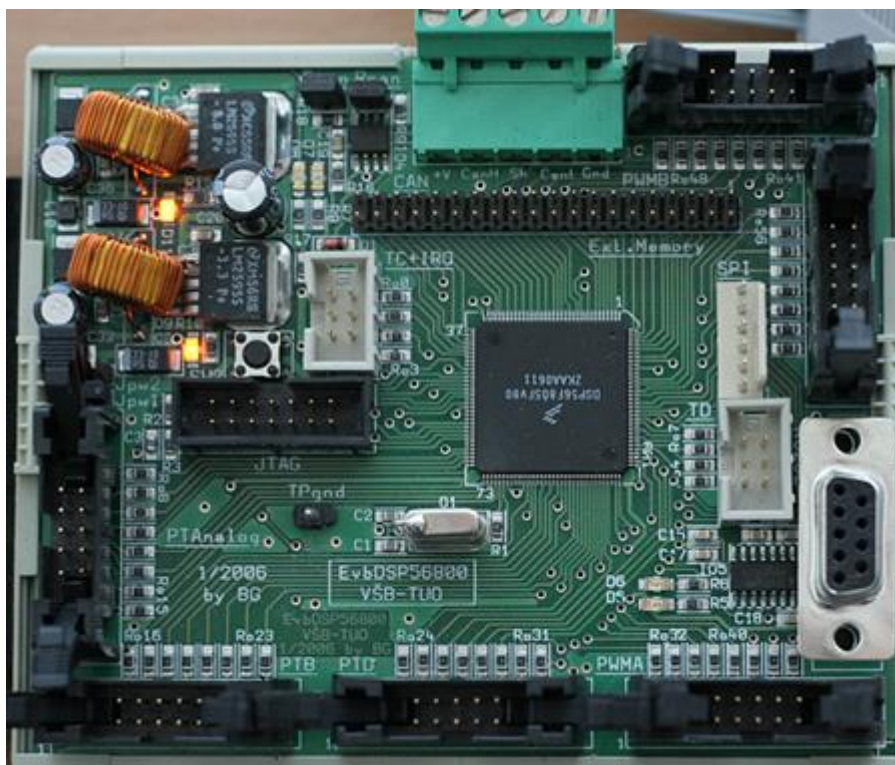
Jedná se o šestnácti - bitový digitální signálový procesor, jehož základní vlastnosti jsou:

- výpočetní výkon až 40 MIPS
- napájecí napětí 3,3V
- 32252 x 16 bitů flash paměti
- 2048 x 16 bitů RAM paměti
- 2 x 6-ti kanálové PWM
- CAN rozhraní
- 14 GPIO bran
- JTAG rozhraní
- 12-ti bitový 8-mi kanálový AD převodník
- 4 x 32 bitové časovače

[6]



Vývojový kit je napájen napětím 12V a obsahuje stabilizátory, které poskytují napětí 3,3V pro digitální signálový procesor a periférie a 5V pro periférie. Všechny periférie jsou vyvedeny na standardní počítačové konektory. Vývojový kit obsahuje také převodník pro komunikaci s PC po sběrnici RS232. Hodinový signál pro DSP zajišťuje krystal 8MHz, který je PLL násobičkou převeden na 80MHz. Na vývojovém kitu je dále LC článek pro filtraci referenčního napětí pro AD převodník.



obr.6.3 vývojový kit s DSP

### **6.3 Výpočet frekvence A/D převodníku**

Frekvence vzorkování a počet vzorků pro výpočet frekvenčního spektra musí být rozumě zvolen. Vzhledem k tomu, že interní funkce vývojového prostředí Processor expert<sup>TM</sup> FFT má konstantní vstupní parametry pro počet vzorku:  $N = 4 \dots 64, 128, 256 \dots 2048$ , byl vybrán počet vzorku  $N = 128$ .

Reakce dynamických obrazců na změnu tvaru frekvenčního spektra jsou dány nízkými frekvencemi, které jsou charakteristické pro basy (25Hz-300Hz). Pokud je žádaný krok ve spektru frekvence  $f_A = 25\text{Hz}$  a  $N = 128$  vzorků, vzorkovací frekvence  $f_s$  se vypočítá takto:

$$f_s = N \cdot f_A = 128 \cdot 25 = 3200\text{Hz}$$

Aby byl splněn Shannon-Kotělníkův teorém, je nutno respektovat spektrum do maximální frekvence:

$$f = \frac{f_v}{2} = \frac{3200}{2} = 1600Hz$$

## **7. Software řídicího systému pro vykreslovací jednotku**

### **7.1 Program v DSP**

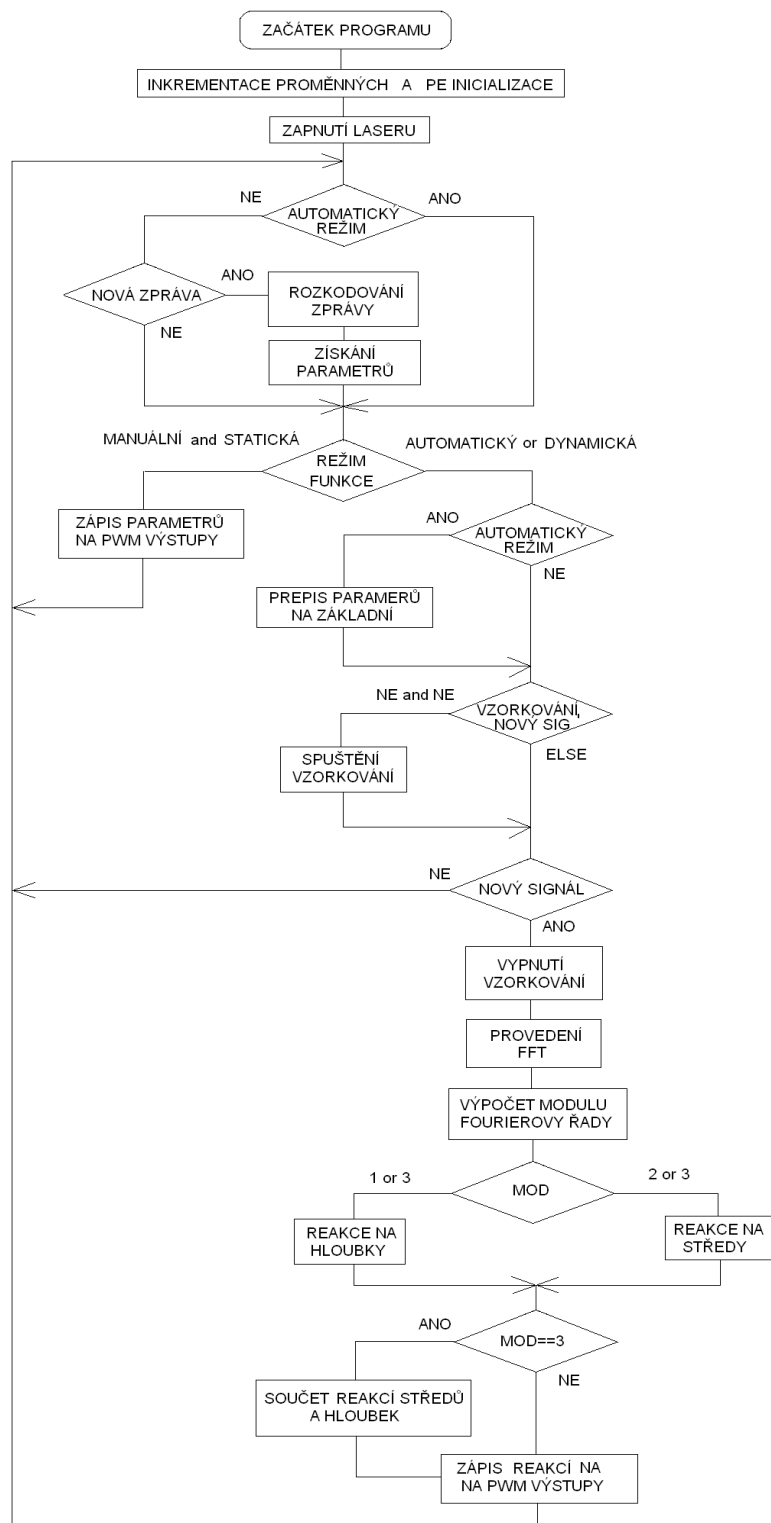
Program v DSP je naprogramován v kódu C v programovacím prostředí Code Warrior, kde je využit nástroj Processor Expert™. Je to nástroj vývojového prostředí pro návrh aplikací v embedded systémech.

Hlavním rysem systému Processor Expert™ je unifikovaný přístup k návrhu, který uživatele odlišuje od hardware daného embedded systému. Tento přístup je založen na použití komponent, tzv. Embedded Beans™, které v mnohém připomínají technologii Java Beans. Processor Expert™ obsahuje základní sadu těchto Embedded Beans, které zapouzdřují funkčnost některého z mikrokontrolérů, které se v embedded systémech používají. Protože tato základní sada je pro všechny podporované procesory v maximální míře unifikovaná, je uživatel při návrhu aplikace ušetřen nutností studovat odlišnosti jednotlivých procesorů. Processor Expert™ obsahuje také znalostní databázi, která obsahuje modely podporovaných procesorů a údaje o nich. Tato databáze se používá ke kontrole správnosti uživatelských požadavků (časování, velikost paměti).

[7]

Soubory do kterých programátor v rámci PE zapisuje kód:

- Hlavní program – je nazván shodně s projektem s koncovkou, obsahuje inkrementaci proměnných a hlavní smyčku programu.
- Events- v tomto souboru je kód, který obsluhuje přerušení, jinak též události, vyvolané komponenty, které programátor přidal do projektu. Tyto metody je nutné co nejrychleji obsloužit z důvodu přerušení chodu vykonávání instrukcí v hlavním programu.



obr.7.1 vývojový diagram programu v DSP

Aby bylo z vývojového diagramu patrné co program dělá a kde se zrovna nachází, popíšeme si pojmy a stavy , ve kterých se program zobrazovací jednotky může nacházet:

- Režim - Automatický (AUTO) nebo Manuální. Pokud jednotka pracuje v manuálním režimu, je schopna přijímat parametry obrazu z PC a provádět statickou i dynamickou funkci. V automatickém režimu vykonává jednotka pouze dynamickou funkci se základními parametry nadefinované v DSP. Komunikace není v tomto režimu možná.
- Funkce – Statická a Dynamická. Funkce statická je pro tvorbu statických obrazců. Funkce dynamická je pro tvorbu dynamických obrazců reagujících na spektrum. Přepnutí funkcí lze provést pouze v manuálním režimu a to pomocí parametru poslaného z PC
- MOD – Jde o druh dynamické funkce. 1.MOD je reakce na hloubky spektra přiváděného signálu (50-200Hz). 2.MOD je reakce na středy (300-1550Hz).
- Nová zpráva – je hodnota typu bool, která má hodnotu 1 pokud je přijato po sériové lince z PC správný počet a druh znaků.
- Vzorkování – je stav kdy program v přerušení načítá a ukládá vzorky signálu přiváděného na analogový port. U tohoto signálu se po uložení 128- mi hodnot provede FFT.
- Nový signál – je proměnná, která slouží k blokaci FFT analýzy a následujících funkcí dokud nemá signál k analýze dostatečný počet vzorků.

#### 7.1.1 Statická funkce

Program s nastavením statické funkce přijímá spočítané parametry z PC (d1, d2, střída) a má následující posloupnost základních instrukcí v nekonečné smyčce:

1. Pokud přijde z PC zpráva, proběhne přepis parametrů pro statickou funkci.
2. Zápis parametrů střídavy pro motor2 a šířky pulsů pro servomotory1,2 na PWM moduly.

Úplnou stabilizaci obrazu lze docílit doladěním potenciometru, ten v malém rozsahu umí měnit otáčky motorem2 a je umístěn na vykreslovací jednotce.

Vstupní parametr funkce, která nastavuje PWM signál pro servomotory jsou šířky pulzů v log1. Je nutné v programu vypočítat tyto hodnoty z parametrů  $d1$ ,  $d2$ , ty představují délku zasunutého táhla do osy rotoru.

Měřením bylo zjištěno, že posunovací zařízení, která ovládají servomotory mají při plném pohybovém rozsahu (vychází z poloměru kružnice, která není ořezaná okraji krabice) zasunutí táhla v ose rotoru  $d = 0 - 2.5\text{mm}$ , rozsah šířky pulzu v log1  $\Delta T_{HIGH} = 0 - 1250\mu\text{S}$ . Je tedy možný přepočet bez desetinné čárky, která by zvětšovala výpočetní čas:

$$T_{HIGH} = d \cdot \frac{1250}{2,5} \Rightarrow T_{HIGH} = d \cdot 500$$

Víme, že krajní nastavení úhlu natočení pro nulové zasunutí táhla do osy rotoru je při  $T_{HIGH} = 2300\mu\text{S}$ . Pro krajní natočení na opačnou stranu, tedy maximální zasunutí táhla do osy rotoru, je při  $T_{HIGH} = 750\mu\text{S}$ . V programu DSP se pohybuje  $d$  v rozsahu 0 - 250, důvodem je vyhnutí desetinné čárky.

Přepočet z  $d$  na  $\Delta T_{HIGH}$  lze vyjádřit pro program v DSP takto:

$$T_{HIGH} = 2300 - \left( d \cdot \frac{1250}{250} \right) \Rightarrow T_{HIGH} = 2300 - (5 \cdot d)$$

Vstupní parametr pro nastavování střidy pro motor2 na portu PWMB je typu „unsigned intriger“, který má rozsah od 0 to 65535. Je proto nutné přepočítat střidu z rozsahem 0-999 na rozsah daného typu. Byl zvolen zkrácený rozsah vstupního parametru funkce „střída“ na 0 až 62000.

Přepočet z střidy na vstupní parametr funkce nastavující PWM signál („unsigned int“):

$$st = \frac{střída}{1000} \cdot 62000 \Rightarrow st = střída \cdot 62$$

Ukázka kódu pro zápis na PWM moduly:

```
if(automatickyRezim==0 & statickaFunkce==1)
{
    strida=st*62;
    PWM2_SetRatio16(strida);
    sirkaImpulzu=2300-(d1*5);
```

```

        PWM1_SetDutyUS(sirkaImpulzu);

        sirkaImpulzu=2300-(d2*5);
        PWM3_SetDutyUS(sirkaImpulzu);
    }

```

Ukázka kódu pro rozšifrování přijaté zprávy z PC a následný přepis parametrů:

```

if(novaZprava==1)
{

    for(i=1;i<=9;i++)//promena pole v byte do pole int
    {
        //nechci porovnávat znaky: "S" a "D" pro výběr režimů
        byte cisloByte=pole[i];
        i--;
        switch (cisloByte)
        {
            case 48:
                pole2[i] = 0;
                break;
            case 49:
                pole2[i] = 1;
                break;
            case 50:
                pole2[i] = 2;
                break;
            case 51:
                pole2[i] = 3;
                break;
            case 52:
                pole2[i] = 4;
                break;
            case 53:
                pole2[i] = 5;
                break;
            case 54:
                pole2[i] = 6;
                break;
            case 55:
                pole2[i] = 7;

```

```

        break;
    case 56:
        pole2[i] = 8;
        break;
    case 57:
        pole2[i] = 9;
        break;
    }
    i++;
}

//vynulovani - aby se mohlo opetne naplnit pole
i = 0;

//ziskani parametu po statickou funkci
if (pole[0]== 83)
{
    statickyRezim = 1; //prepnuti rezimu
    stovka = pole2[0] * 100;
    desitka = pole2[1] * 10;
    jednotka = pole2[2];
    d1 = stovka + desitka + jednotka;

    stovka = pole2[3] * 100;
    desitka = pole2[4] * 10;
    jednotka = pole2[5];
    d2 = stovka + desitka + jednotka;

    stovka = pole2[6] * 100;
    desitka = pole2[7] * 10;
    jednotka = pole2[8];
    st = stovka + desitka + jednotka;
}
}

```

### 7.1.2 Dynamická funkce

Zvolený způsob vykreslování laserem není vhodný pro zobrazení amplitudového spektra jako takového, avšak můžeme přenést některé jeho vlastnosti takřka v reálném čase do parametrů promítaného obrazce. Jestli se budou výrazně měnit dominantní harmonické složky vstupní signálu, jsme schopni vhodným algoritmem měnit parametry obrazu, tak aby změny ve spektru, které jsou lidským uchem rozpoznatelné, byly také postřehnutelné na reagujícím laserem vykreslovaném obrazci.

Vhodný vstupní signál s výraznými změnami amplitudovým spektrem v čase v pásmu 50-1600Hz je hudba. Akční členy rozmítacího zrcátkového systému, které jsou schopny správně a rychle reagovat na dynamické změny jsou servomotory. Ty mění poloměr  $R$  vnější kružnice  $K$  a vzdálenost  $d$  geometrického systému. Otáčky motorů lze při reakci na vstupní signál libovolně měnit a vytvářet tak různé typy obrazců. Otáčky však nebudou mít žádnou vazbu na spektrum signálu.

Algoritmus, který byl vytvořen pro získání parametrů obrazu se dá rozdělit do dvou částí. Těmi jsou reakce na amplitudové spektrum v oblasti hloubek (50-200Hz) a reakce na spektrum v oblasti středů (300-1550Hz). Rytmus v moderní hudbě tvoří převážně hloubky, mají v tomto programu dle velikosti amplitudy měnit velikost obrazce. Podle frekvence největší amplitudy harmonické složky v oblasti středů, které mají v hudbě zastoupení hlavně ve zpěvu nebo zvuku kytary, se obraz zaobluje tím víc, čím vyšší dominantní frekvence daného pásma je.

Program v dynamické funkci lze rozdělit na tyto blokové operace jdoucí po sobě ve smyčce:

#### 1. Vzorkování signálu

Vzorkovací frekvenci jsme si již odvodili v kapitole Hardware. Časovou základnu pro opakované měření (A/D převodník) dle určené frekvence tvoří časovač, který vyvolává přerušení periody rychlostí vzorkovací frekvence. Uložení do pole signálu se provádí ve funkci, která je přerušením A/D převodníku při ukončení měření. Ačkoli je A/D převodník 12-ti bitový, jedna z funkcí v PE převede změřenou hodnotu do rozsahu „short“ což je 16-ti bitový „intriger“ v rozsahu -32768 až 32767. Získané vzorky se ukládají do pole komplexního typu CFrac16. Jedná se o podobný datový typ s „int16“ („short“), ale procesor u něj provádí posun do hodnot v intervalu  $< -1; 1 >$ , tím je zabráněno přetečení. CFrac16 obsahuje navíc reálnou a imaginární složku.

Ukázka kódu pro vzorkování signálu ze souboru Events:

```
void AD1_OnEnd(void)
{
    AD1_GetChanValue(0, &ADvalue); //16-bitove cislo

    pX[i].real = ADvalue;
    pX[i].imag = 0;

    AD1_Stop();

    i++;
}
```



```

        if(i==128)
        {
            // zastavení AD převodníku
            TI1_Disable(); // vypnutí triggeru převodníku
            i=0;
            novySignal = 1;
            vzorkovani=0;
        }
    }
}

```

## 2. Frekvenční analýza

Funkce FFT, která se vykonává v dynamické funkci je dostupná v nástroji Processor Expert™ se vstupními parametry typu CFrac16 a počtem prvků..

Pokud dostaneme spektrum v komplexním tvaru, je potřeba vypočítat jejich absolutní hodnoty, abychom dostali amplitudové spektrum, s kterým budeme dále nakládat.

Bylo zjištěno , že amplitudové spektrum v programu DSP dosahuje jedné pětiny velikosti svého rozsahu. To je způsobeno malým vstupním napětím maximálně do jednoho voltu. Napěťový rozsah A/D převodníku je přitom do 3,3V. Maximální vstupní napětí 1V bylo stanoveno, aby se neořezával signál v záporné půlvlně signálu.

### Ukázka kódu pro vypočtení amplitudového spektra

```

res=dfr16CFFT (pCFFT, pX, pZ); // provedení FFT

for(i=0;i<(N/2)+1;i++)
{
    // výpočet absolutních hodnot z komplexních čísel
    pY[i].real=mfr16Sqrt(add((mult(pZ[i].imag,pZ[i].imag)),
        (mult(pZ[i].real,pZ[i].real))));

    pY[i].real=5*pY[i].real;
    pY[i].imag=0;
}

```

### 3. Reakce obrazu na hloubky daného zvukového spektra (MOD1):

U této reakce obrazu na určitý rozsah frekvencí je potřeba zjistit nejvyšší amplitudu v dané frekvenční oblasti 50-300Hz. Tu převést dle rozsahu amplitudy pro kladné hodnoty amplitudového spektra ( $pY[i] = \langle 0 ; 32767 \rangle$ ) na vychylka1 a vychylka2 s rozsahem  $\langle 0 ; 250 \rangle$ . Výchylka zde představuje posun táhla do osy rotoru.

Výpočet odchylky:

$$\text{odchylka} = pY[i] \cdot \frac{250}{32767} \Rightarrow \text{odchylka} \cong \frac{pY[i]}{132}$$

Odchylky jsou pro oba motory stejné, protože tak se dosáhne změna velikosti obrazce. Zanedbávají se rozdílné délky ramen na rotorech A1, A2. Parametry  $R$  a  $d$ , vycházející z geometrického systému ovalujících se kružnic se zvětšují současně o stejnou konstantu, jsou tedy ve stejném poměru.

Nesmí se zapomenout na vynulování nejvyšší nalezené amplitudy před každým novým hledáním v nekonečném cyklu

Ukázka kódu v pro získání odchylky v modu1:

```
for (i = 1; i <= 7; i++)//index 1-7 -> 50-200hz
{
    if (maxAmpBasy < pY[i].real)
        maxAmpBasy = pY[i].real;
}
//prepocet na parametry tmax pwm 32767/132=248[mm*100]
vychylkaD1 = maxAmpBasy / 132;
vychylkaD2 = maxAmpBasy / 132;

//prepocet pro mod3 tmax pwm (32767/132)/2=124[mm*100]
vychylkaD1B = maxAmpBasy / 116;
```

### 4. Reakce obrazu na středy daného frekvenčního spektra (MOD2):

U této reakce obrazu na určitý rozsah frekvencí je potřeba zjistit nejvyšší amplitudu v dané frekvenční oblasti 300 - 1550Hz a převést ji dle rozsahu indexů frekvence (11 - 61) na výchylku táhla spojeného se ramenem zrcátka s rozsahem (0 – 250). Výchylka zde představuje posun táhla do osy rotoru a lze ji odvodit:

$$výchylkaD = pY[i] \cdot \frac{250}{61-11} \Rightarrow \text{odchylka} = pY[i] \cdot 5$$

Reakce se děje pouze na servomotoru1, protože tak docílíme zaoblení obrazce. Velikost zaoblení určovat frekvence s nevyšší frekvencí amplitudy snímaného spektra.

Ukázka kódu v pro získání odchylky v modu2:

```
for (i = 11; i <= 61; i++)
{
    if (maxAmpStredy < pY[i].real)
    {
        maxAmpStredy = pY[i].real;
        maxIndexStredy = i;
    }
}

//prepocet na tmax pwm 5*( maxIndexStredy -11)=250 [mm*100]
vychylkaD1 = 5 * (maxIndexStredy - 11);
vychylkaD2 = 0

//prepocet pro mod3,tmax pwm 5*(maxIndexStredy-11)/2=125 [mm*100]
vychylkaD1S = (5 * (maxIndexStredy - 11)) / 2;
```

##### 5. Reakce obrazu na hloubky i středy frekvenčního spektra (MOD3):

V tomto modu se použijí odchylky vypočítané v modech 1,2 které jsou polovinami reakci předchozích modů. Pokud tyto reakce sečteme přizpůsobíme tak rozsah odchylky na maximum. Zároveň je zaručeno, že výsledek nepřesáhne maximální povolenou hodnotu, vycházející z vykreslení neořezaných kružnic vykreslovací jednotkou.

##### 6. Zápis reakcí na PWM výstupy na portu PWMA

Typy parametrů přijímané z PC v dynamické funkci byly zvoleny tak, aby se dal ovlivňovat základní i dynamický obrazec, který se vlivem odchylek zvětšuje nebo zaobluje. Těmito parametry jsou:

- Gain – je násobitelem odchylky v rozsahu 0,1 – 1. Ovlivňuje dynamický rozsah, a tudíž maximální velikost obrazce při reakci na amplitudové spektrum.

- DefaultD1 – Je základní posun servomotoru1 v jeho rozsahu 10-80%
- DefaultD2 – Je základní posun servomotoru2 v jeho rozsahu 20-80%
- Mod – Podle typu modu (1, 2, 3) probíhá daná reakce na spektrum signálu
- stD – Je to střída v rozsahu 0 – 999, v dynamické funkci má stejné opodstatnění jako ve statické funkci. Nastavuje se podle Základní velikosti  $d$  (DefaultD2 ).

Při zápisu na PWM periferie je třeba zahrnout proměnné ovlivňující obraz. Je potřeba zavést vztah, podle kterého se budou výchylky posunovacích zařízení s ohledem na základní zobrazovaný obraz počítat na  $\Delta T_{HIGH}$ , jehož dosazení do funkce PWM periferie už jsme si znázornili v podkapitole kapitole Statická funkce.

$$\Delta T_{HIGH} = gain \cdot odchylkaD \cdot \left[ \frac{1250 - \left( defaultD \cdot \frac{1250}{250} \right)}{250} \right] + \frac{defaultD \cdot 1250}{250}$$

$$\Delta T_{HIGH} = gain \cdot odchylkaD \cdot \left( \frac{1250}{250} - \frac{\frac{defaultD \cdot 1250}{250}}{250} \right) + 5 \cdot defaultD$$

$$\Delta T_{HIGH} = gain \cdot odchylkaD \cdot \left( 5 - \frac{defaultD \cdot 1250}{250^2} \right) + 5 \cdot defaultD$$

$$\Delta T_{HIGH} = gain \cdot odchylkaD \cdot (5 - defaultD \cdot 0,02) + 5 \cdot defaultD$$

Ukázka kódu v pro zápis odchylek na PWM výstupy:

```
strida=stD*62;

PWM2_SetRatio16(strida);

DELt1 = gain * (double)vychylkaD1 * (5.0 -
(double)defaultD1 * 0.02) + 5 * defaultD1;

sirkaImpulzu=2300-((unsigned int)DELt1);
PWM1_SetDutyUS(sirkaImpulzu);

DELt2 = gain * (double)vychylkaD2 * (5.0 -
(double)defaultD1 * 0.02) + 5 * defaultD2;

sirkaImpulzu=2300-((unsigned int)DELt2);

PWM3_SetDutyUS(sirkaImpulzu);
```

Protože jsme se zde nemohli vyvarovat práci s desetinnou tečkou, musí se  $\Delta T_{\text{HIGH}}$  konvertovat do typů s plovoucí desetinnou tečkou. Po výpočtu se tato proměnná konvertuje do typu „unsigned int“, protože je tento typ požadovaný jako vstupní parametr funkce nastavující PWM signál.

## **7.2 Program v PC**

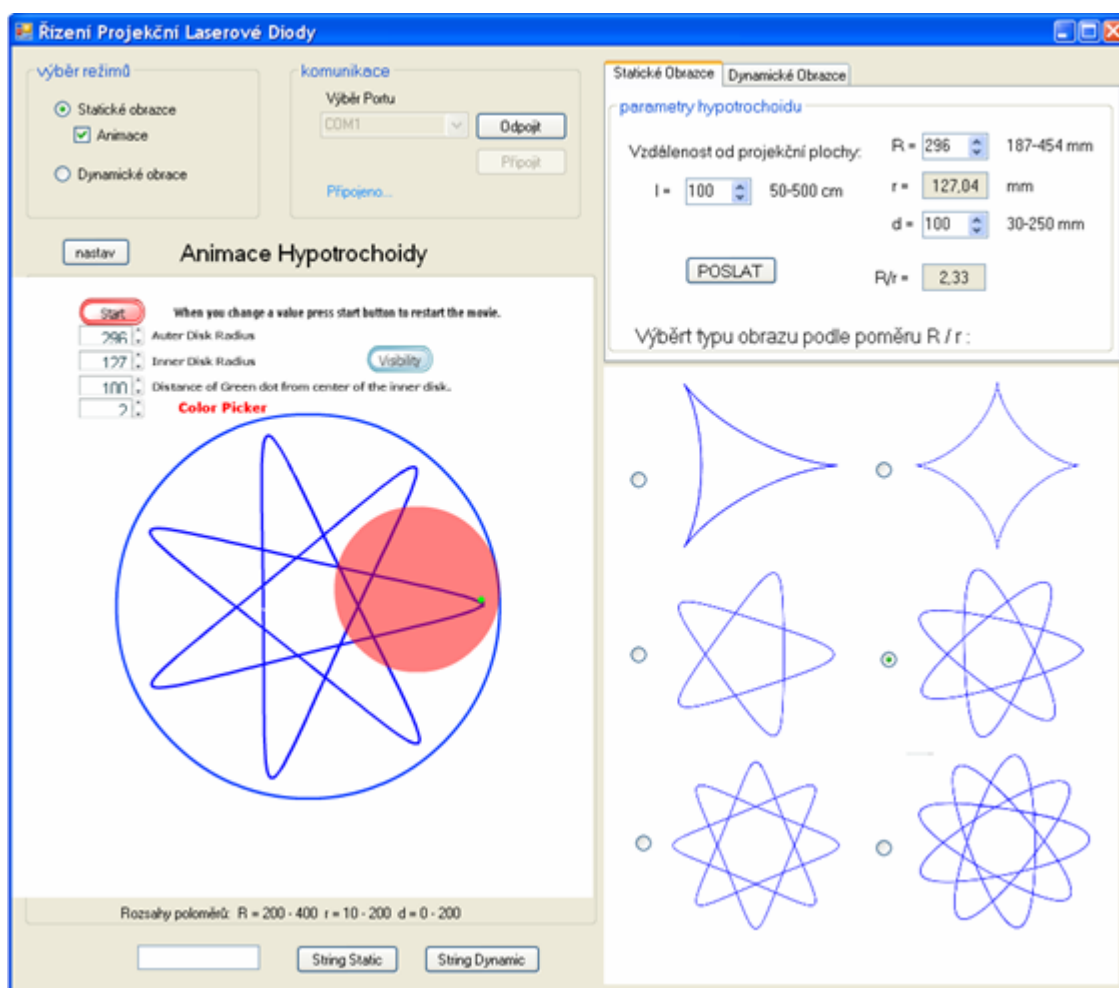
V PC je aplikace naprogramovaná v prostředí Microsoft Visual Studio v kódu C#. Jedná se o typ aplikace Windows, ve které se poměrně snadno vytváří interface s klasickými grafickými prvky. Tato aplikace počítá parametry z geometrického systému ovalující se vnitřní kružnice na parametry vykreslovací jednotky.

V FLASH animaci, která je součástí přílohy III, se zadávají a vykresluji parametry ovalujících se kružnic. Pro test se statickými obrazy můžeme provést porovnání vykreslených obrazců v animaci s vykreslenými obrazy vytvořenými pomocí zrcátkového systému.

Všechny vztahy, které se v této kapitole objevují v kódech a nejsou odvozeny jsou vyjádřeny v 4. kapitole.

### 7.1.2 Statická funkce

V programu jehož znázornění je na obr.7.1 si uživatel volí parametry, které pošle tlačítkem POSLAT. Aplikace podle nastavených hodnot poskládá „strink“, který pošle po sériové lince do DSP. Parametry, které lze nastavit u statické funkce jsou: vzdálenost od promítací plochy, velikost statické kružnice, poměr velikosti kosti kružnic. Z těchto parametrů se dopočítávají pro vizualizaci poloměr  $r$  kružnice  $k$  a možné rozsahy. Pro DSP se počítají veličiny  $d1, d2$  a určuje střída pro motor2.



Obr.7.1 ukázka interface v statické funkci, zapnutá animace

Okno programu je rozděleno do skupin (panelů):

- Komunikace – Pro zprovoznění komunikace je potřeba zvolit port a připojit se k němu. Panel též obsahuje tlačítko pro odpojení a status.
- Výběr režimů – Při výběru statické obrazce je povoleno zobrazit animaci. Při přepínání režimů se obnovuje funkčnost ovládacích prvků v panelu nastavení.
- Animace – U statické funkci a zadání shodných parametrů z panelu nastavení kreslí vykreslovací laserový systém shodný obrazec.
- Nastavení – Lze zvolit nastavení pro statické obrazce přepnutím záložky. V tomto panelu jde nastavit velikost poloměru  $R$  statické vnější kružnice, vzdálenost vykreslovacího bodu od středu vnitřní kružnice  $k$  a vzdálenost vykreslovací jednotky od promítacího plátna.
- Výběr obrazce – Při změně typu zobrazovaného obrazce se dopočítávají potřebné veličiny jako jsou rozsahy proměnných a poloměr  $r$  vnitřní kružnice
- Kontrola zprávy – pod animací jsou prvky pro kontrolu zprávy pro DSP která odpovídá aktuálnímu nastavení parametrů.

část ukázkové metody „GetZpravaVlaserParam“:

Kód, který přepočítává parametry pomocí vztahů uvedených v 4. kapitole. Metoda vrací „string“ který obsahuje parametry hypotrochoidu v zrcátkovém systému. Prve jsou vstupní parametry převedeny do typů s desetinnou čárkou a poté použity k výpočtům  $d_1$  a  $d_2$ , ty jsou pak zaokrouhleny.

```
Public static string GetZpravaVlaserParam(int l, int R, int r, bool
statRez, int d, double pomer)
{
    //-----urceni indexu pole, d1, d2, strida-----

    double Rd=Convert.ToDoubleZPRÁVY;
    double rd=Convert.ToDoublezprávy;
    double dd=Convert.ToDouble(d);
```

```

double ld=Convert.ToDouble(l)*10;

double d2d = ((Rd - rd) * 15.3) / (2 * ld);
double d1d = (dd * 16.5) / (2 * (ld + 40));

d2d=Math.Round(d2d, 2);
d1d=Math.Round(d1d, 2);

//prepočet vzdalenosti na index v poli stríd podle d2
double pom = (d2d * 10)-1;
pom=Math.Round(pom, 0);
int inxSloupec = Convert.ToInt16(pom);

int inxRadek = 3;
//určení idexu správného řádku stříd podle pomeru R/r
if (pomer == 7) inxRadek = 0;
if (pomer == 4) inxRadek = 1;
if (pomer == 3) inxRadek = 2;
    if (pomer == 2.66) inxRadek = 3;
    if (pomer == 2.5) inxRadek = 4;
    if (pomer == 2.33) inxRadek = 5;
    if (pomer == 2.25) inxRadek = 6;

//zapsání strávné střídý z pole do proměnné
int strida = stridy[inxRadek, inxSloupec];
.
.
.

```

část ukázkové metody „GetZpravaVlaserParam“:

Dokončení metody pro získání řetězce znaků s parametry pro hypotrochoidu.

Při vytvoření návratové hodnoty „string“ je nutné aby měl vždy stejný počet symbolů.

```

.
.
.
//-----vytvoreni stringu pro d1,d2 a stridu-----

//d1 a d2 se posílají v mm*100

int dlz = Convert.ToInt16(d1d * 100);
int d2z = Convert.ToInt16(d2d * 100);
string stridas = strida.ToString();

```



```

int stovky=d1z/100;
int desitky=(d1z%100)/10;
int jednotky = d1z%10;
string d1s = stovky.ToString()+desitky.ToString()+jednotky.ToString();

stovky = d2z/100;
desitky = (d2z % 100) / 10;
jednotky = d2z%10;
string d2s = stovky.ToString()+desitky.ToString()+jednotky.ToString();

//--vytvoreni stringu pro vyber rezimu staticky/dynamicky-
string rezim="C";
if (statRez == true) rezim = „S“;

else rezim = „D“;

//---poskladani stringu a vytvoreni zpravy pro poslani do DSP-----
// @ zavináč je startovacím znakem pro příjem dat v DSP
string zprava = „@“ + rezim + d1s + d2s + stridas;
return zprava;

}

```

Kód pro ošetření stisku tlačítka POSLAT

[14]

```

private void bt_send_Click(object sender, EventArgs e)
{
    try
    {
        /* Posleme retezec na seriovou linku */
        if (statickeObrazce == true)

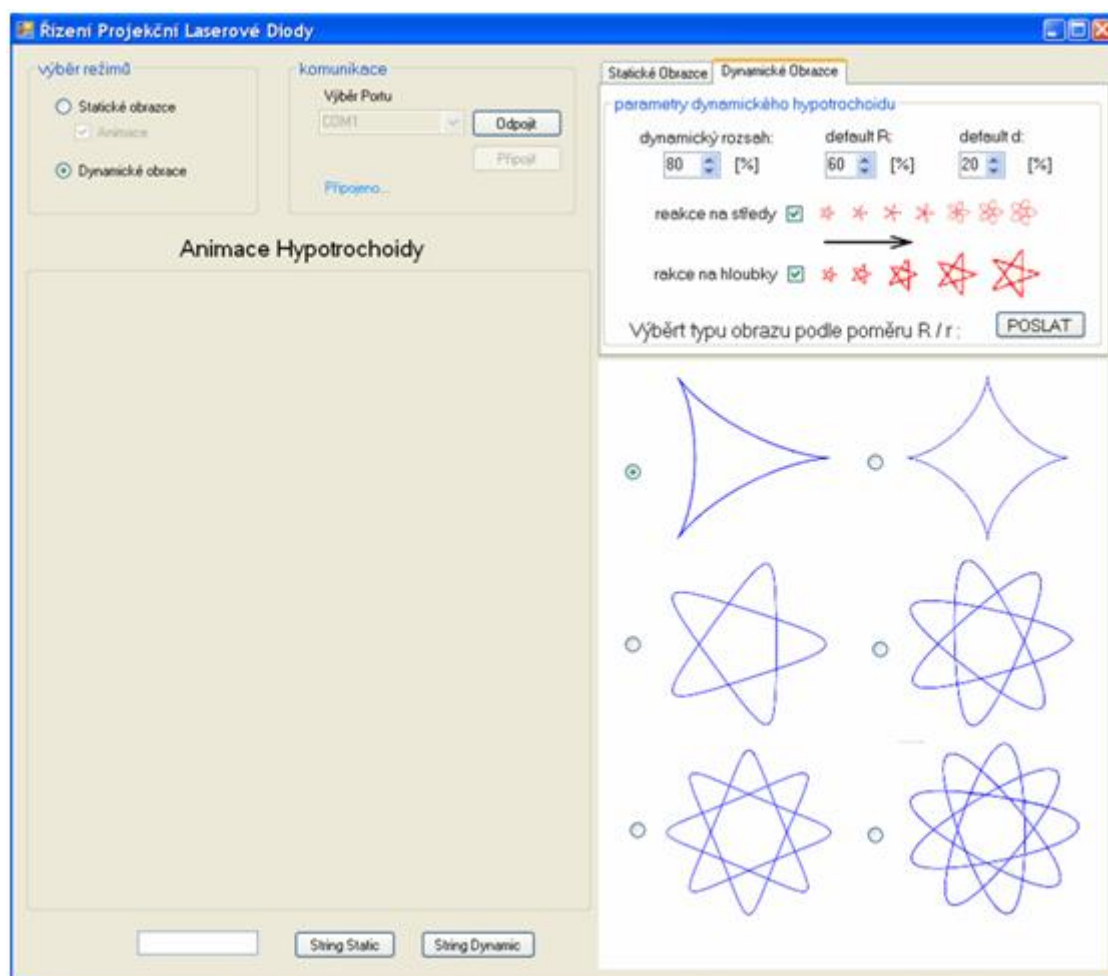
            zprava =
statickeOb_funkce.GetZpravaVlaserParam(vzdalenost_l,polomer_R,polomer_
r,statickeObrazce, vzdalenost_d,pomer);

        serialPort1.Write(zprava);
    }
    catch (Exception)
    {
        /* V pripade nezdaru vypiseme chybove hlaseni */
        stat_label.Text = „neznámá chyba při odesílání“;
    }
}

```

### 7.1.2 Dynamická funkce

V programu jehož znázornění je na obr si uživatel volí parametry dynamických obrazců, které pošle tlačítkem POSLAT. Aplikace podle nastavených hodnot poskládá „string“, který pošle po sériové lince do DSP. Parametry, které lze nastavit u dynamické funkce jsou: dynamický rozsah, základní velikosti dynamického obrazce, typ reakce na frekvenční spektrum. Z těchto parametrů se dopočítávají pro DSP veličiny  $d_1, d_2$  a určuje se střída pro motor2.



Obr.7.2 ukázka interface v dynamické funkci

Okno programu pro dynamické obrazce má podobné rozdělení panelů jako u statických obrazců, rozdíl tvoří tyto prvky:

- Animace – U dynamické funkce nelze zobrazit, protože animace odpovídá statickým obrazcům, které nereagují na vnější podněty.
- Nastavení – Lze zvolit nastavení pro dynamické obrazce přepnutím záložky. V tomto panelu jdou nastavit základní velikosti dynamického obrazce, kterými jsou: poloměr  $R$  statické kružnice  $K$  a vzdálenost  $d$  vykreslovacího bodu od středu vnitřní kružnice  $k$ . Dále je možnost zvolit si mezi typy reakcí na frekvenční spektrum. Nastavená procenta odpovídají celkovému rozsahu.

část ukázkové metody „GetD1“:

Metoda, která je součástí třídy `dynamickeOb_funkce` vrací hodnotu pro posun táhla servomotoru1 v DSP. Hodnota odpovídá celkovému rozsahu posunovacího zařízení, a tudíž i celkovému rozsahu vzdálenosti  $d$  geometrického systému.

```
class dynamickeOb_funkce: statickeOb_funkce
{
    public static int GetD1(int dlproc)
    {
        double pom;
        pom = (dlproc * 25) / 100;
        pom = Math.Round(pom, 0);

        int pom2 = Convert.ToInt16(pom);
        return pom2;
    }
    .
    .
    .
}
```

Část ukázkové metody „GetZprava“ pro dynamickou funkci:

Tato metoda třídy `dynamickeOb_funkce` vrací řetězec znaků, který se posílá do DSP přes seriovou linku. Tento „String“ tvoří gain-dynamický rozsah, mod - typ reakce na frekvenční

spektrum, základní posuny servomotorů a střidu pro motor 2, její zvolení však v kódové ukázce není zahrnuto. Zde je důležité aby zpráva měla vždy stejný počet znaků.

```
public static string GetZprava(int defD1, int defD2, int mod, int
gain, double pomer)
{
    //----vytvoreni stringu pro defD1,defD2,gainu a modu-----
    string gainS=((gain/10)-1).ToString();
    string modS = mod.ToString();
    //int stovky = defD1 / 100;
    int desitky = defD1 / 10;
    int jedtotky = defD1 % 10;
    string defD1s = desitky.ToString() + jedtotky.ToString();

    //stovky = defD2 / 100;
    desitky = defD2 / 10;
    jedtotky = defD2 % 10;
    string defD2s = desitky.ToString() + jedtotky.ToString();
    .
    .
    .
}
```

## **8. Závěr**

Zhodnocení dosažených výsledků pro statické obrazce je provedeno pomocí animačního programu, který je součástí přílohy. Obrazec, který vznikne animací s určitými parametry bereme jako teoretický předpoklad pro obrazec vykreslený pomocí zrcátkového rozmítacího systému. Fotografie obrazce vykresleného laserem jsou tmavé, protože musela být nastavená vysoká doba závěrky fotoaparátu, aby byl vykreslený obraz celistvý.

Pro zhodnocení dynamických obrazců reagujících na frekvenční spektrum signálu je natočeno video s titulky a zvukovou nahrávkou (hudbou), které je součástí přílohy IV. Byly též zhotoveny fotografie dynamických obrazců s reakcí v modu2 na periodický signál s konstantní harmonickou frekvencí.

### **8.1 Test se statickými obrazci**

Srovnání výsledků animace a vykresleného obrazce se stejnými parametry:

- Parametry hypotrochoidy:  $R = 300\text{mm}$   $r = 75\text{mm}$   $d = 50\text{mm}$

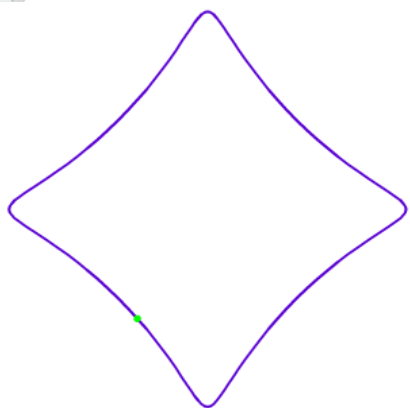
Start When you change a value press start button to restart the movie.

Auter Disk Radius: 300

Inner Disk Radius: 75

Distance of Green dot from center of the inner disk: 50

Color Picker: 1



obr.8.1 animace  $k = 4$



obr.8.2 fotografie zobrazeného obrazce,  $k = 4$

- Parametry hypotrochoidy:  $R = 250\text{mm}$   $r = 100\text{mm}$   $d = 60\text{mm}$

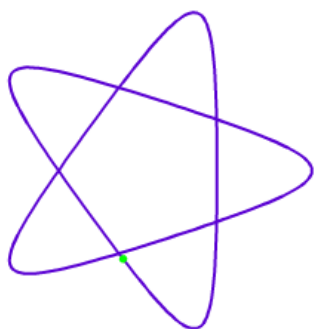
Start When you change a value press start button to restart the movie.

Auter Disk Radius: 250

Inner Disk Radius: 100

Distance of Green dot from center of the inner disk: 60

Color Picker: 1



obr.8.3 animace  $k = 2,5$



obr.8.4 fotografie zobrazeného obrazce,  $k = 2,5$

- Parametry hypotrochoidy:  $R = 300\text{mm}$   $r = 113\text{mm}$   $d = 50\text{mm}$

Start When you change a value press start button to restart the movie.

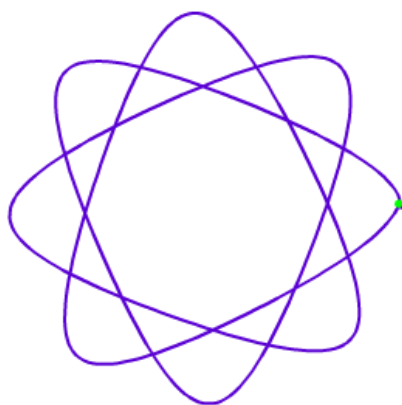
300 Auter Disk Radius

113 Inner Disk Radius

50 Distance of Green dot from center of the inner disk.

1 Color Picker

Visibility



obr.8.5 animace  $k = 2,66$



obr.8.6 fotografie zobrazeného obrazce,  $k = 2,66$

- Parametry hypotrochoidy:  $R = 200\text{mm}$   $r = 89\text{mm}$   $d = 150\text{mm}$

Start When you change a value press start button to restart the movie.

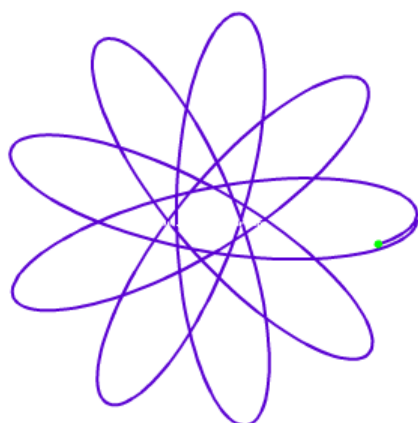
200 Auter Disk Radius

89 Inner Disk Radius

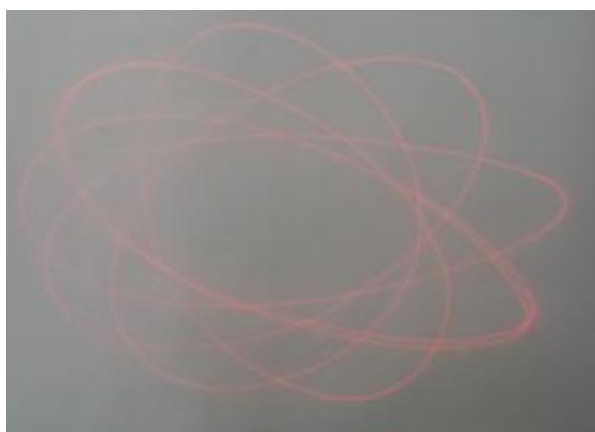
150 Distance of Green dot from center of the inner disk.

1 Color Picker

Visibility

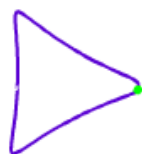
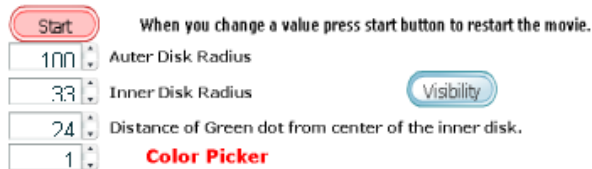


obr.8.7 animace  $k = 2,25$



obr.8.8 fotografie zobrazeného obrazce,  $k = 2,25$

- Parametry hypotrochoidy:  $R = 250\text{mm}$   $r = 100\text{mm}$   $d = 60\text{mm}$



obr.8.9 animace  $k = 3$



obr.8.10 fotografie zobrazeného obrazce,  $k = 3$

### 8.1.1 Zhodnocení výsledků statických obrazců

Podařilo se vykreslit obrazce podle teoretických předpokladů s viditelnými odchylkami. Projevuje se nepřesnost mechanického charakteru, která má také vliv na zkroucení obrazu. Tyto odchylky jsou patrné při vykreslování obrazu s malými hodnotami pro parametry  $R$  a  $d$ . Důvodem jsou malé velikosti posuny táhla ramene v osách rotorů, které se pohybují v řádech desetin milimetrů. Tento problém má částečné řešení, a to přičtením časové konstanty k hodnotě  $\Delta T_{\text{HIGH}}$ , která nastavuje šířku PWM signálu. Tímto přičtením by se provedla korekce na jednom ze servomotorů a dospěli bychom k vykreslení věrohodnějšího obrazce v porovnání s teoretickými předpoklady.

Další nedokonalost, která nejde na fotografiích vidět je to, že obraz rotuje, protože tabulka, podle které se určuje střída pro motor2 nedostačuje k úplné stabilizaci točení. Nicméně je možno rotaci zastavit potenciometrem, který je součástí vykreslovací jednotky. V příloze IV. je na videu zachycena změna typů obrazců s různým poměrem poloměrů kružnic  $k = R / r$ , kde lze sledovat zmíněnou rotaci.

## 8.2 Test s dynamickými obrazci

- Reakce dynamického obrazu na střední pásmo frekvencí 300-1550Hz (MOD=2)

Parametry jsou u tohoto testu nastaveny tak aby byl viditelný rozdíl mezi obrazci podle přiváděné frekvence. Přiváděný signál byl od 400Hz do 1700Hz



*obr.8.11 signál 300Hz*



*obr.8.12 signál 700Hz*



*obr.8.13 signál 1100Hz*



*obr.8.14 signál 1500Hz*

### 8.1.1 Zhodnocení výsledků dynamických obrazců

Test, který je v této práci znázorněn proběhnul a byl natočen se na video, z videa se následně vytvořily fotografie pro různé frekvence. Systém reaguje dle očekávání a zaobluje obraz podle dominantní frekvence v pásmu frekvenčního spektra. Vadou na této funkci je to, že obraz se zaobluje trhaně namísto plynule, jak by tomu mělo být. Důvodem je nedobře vypadající vzorek signálu pro FFT. Změřený vstupní signál by měl začínat a končit v nule, pokud tomu tak není frekvenční spektru je vypočítáno nepřesně. To má za následek určení rozdílné dominantní frekvence pro jiný časový okamžik při konstantní frekvenci na analogovém vstupu.



Řešením by zcela určitě bylo násobení signálu při vzorkování okenní funkcí, to se však nepovedlo naprogramovat. Pravděpodobně z důvodu nedostatečné RAM paměti procesoru program nefungoval správně.

Test reakcí dynamického obrazce na hloubky frekvenčního spektra 50-200Hz a test dynamického obrazce současně na obě pásma proběhnul v laboratoři a byl natočen na video, které je součástí přílohy IV.

V reakci systému na hudbu se výše zmíněné nedostatky systému neprojevují, nebo alespoň nejdou vidět. Odezva systému je zřejmá a přímo reaguje na hudbu s maximální odezvou 65mS. To bylo změřeno pomocí osciloskopu tak, že se ve hlavní smyčce negoval výstupní bit. Výsledná perioda obdélníkového signálu byla dvojnásobkem periody cyklu.

## **9. Literatura:**

- [1] NEVŘIVA, P.: *Analýza signálů a soustav*. Praha: BEN, 2000. 671s. ISBN 80-7300-004-0.
- [2] Palacký, Petr. Neborák, Ivo. *Aplikace signálových procesorů v řízení pohonné jednotky elektromobilu Tatra Beta*. [online], VŠB – TU, c 2007, [cit. 2010-04-20]. URL: <<http://www.roznovskastredni.cz/dwnl/pel2007/01/Palacky.pdf>>.
- [3] Macháček, Z. *Úvod do problematiky digitálních signálových procesorů – DSP*. Studijní texty. VŠB – TU. URL: <[http://www.fei1.vsb.cz/stud\\_mat/K450/](http://www.fei1.vsb.cz/stud_mat/K450/)>.
- [4] Růžička, P. *Ventilátory používané v počítačích – DC brushless Fans*. [online], Internetový blog, c 2003, [cit. 2010-03-28]. URL: <<http://pavouk.org/hw/fan/index.html>>.
- [5] Katalogové listy. *FLASER5MW*. [online], GM Electronic, spol. [cit. 2010-03-28]. URL: <<http://www.gme.cz>>.
- [6] Katalogové listy. *DSP56f805*. [online], Freescale semiconductor. [cit. 2010-04-27]. URL: <<http://www.freescale.com>>.
- [7] *Vývojové prostředí Processor Expert™*. Anotace k přednášce. [online], ČVUT. [cit. 2010-05-5]. URL: <[http://dce.felk.cvut.cz/cak/abstrakty/abstrakt\\_00\\_12\\_08.htm](http://dce.felk.cvut.cz/cak/abstrakty/abstrakt_00_12_08.htm)>.
- [8] *epoitrochoid*. [online], Wikipedia The free Encyklopedia. [cit. 2010-05-5].

URL: <<http://en.wikipedia.org/wiki/File:EpitrochoidIn3.gif>>.

- [9] *hypotrochoid*. [online], Linux Gazette. [cit. 2010-05-5].  
URL: <<http://linuxgazette.net/133/misc/luana/fig-spiro01.png>>.
- [10] Encyklopedia Nation Master. *Spirograph*. Článek.[online], State Master.  
[cit. 2010-05-5]. URL: <<http://www.statemaster.com/encyclopedia/Spirograph>>.
- [11] *obr.3.14 Spirograf*. [online], [cit. 2010-05-5]. URL: <<http://andsol.blox.pl/resource/spir1.jpg>>.
- [12] Dokumentace. *INO –LAB HGD-201HB*. [online], Ino-Lab-RC models. c 2010,  
[cit. 2010-03-28]. URL: <<http://www.inolab.co.kr>>.
- [13] Lukáš,M. *Laserové diody – Část 2. Funkce a základní parametry*. Článek. [online],  
FCC Public. c 2005, [cit. 2010-04-15]. URL: <[http://www.odbornecasopisy.cz/index.php?id\\_document=34933](http://www.odbornecasopisy.cz/index.php?id_document=34933)>.
- [14] *Komunikace po RS 232 (seriovém portu) v C#*. [online], Embos. c 2010,  
[cit. 2010-05-05]. URL: <<http://www.embos.cz/clanky/rozhrani-protokoly/komunikace-po-rs-232-seriovem-portu-v-c>>.

## **10. Přílohy:**

Příloha I. – zdrojové kódy programů pro DSP a PC

Příloha II. – datasheety a katalogové listy DSP

Příloha III. – animace Hypotrochoid

Příloha IV. – video a fotografie z testování

Příloha V. – fotografie přípravku